frontiers in Communications and Networks

ORIGINAL RESEARCH
published: 07 October 2021
doi: 10.3389/frcmn.2021.739414

Entropy-Driven Stochastic Federated Learning in Non-IID 6G Edge-RAN

Brahim Aamer[1]*, *Hatim Chergui*[2], *Mustapha Benjillali*[1] and *Christos Verikoukis*[2]

[1]National Institute of Posts and Telecommunications (INPT), Rabat, Morocco, [2]Centre Tecnologic De Telecomunicacions De Catalunya, Barcelona, Spain

Scalable and sustainable AI-driven analytics are necessary to enable large-scale and heterogeneous service deployment in sixth-generation (6G) ultra-dense networks. This implies that the exchange of raw monitoring data should be minimized across the network by bringing the analysis functions closer to the data collection points. While federated learning (FL) is an efficient tool to implement such a decentralized strategy, real networks are generally characterized by time- and space-varying traffic patterns and channel conditions, making thereby the data collected in different points non independent and identically distributed (non-IID), which is challenging for FL. To sidestep this issue, we first introduce a new *a priori* metric that we call *dataset entropy*, whose role is to capture the distribution, the quantity of information, the unbalanced structure and the "non-IIDness" of a dataset independently of the models. This *a priori* entropy is calculated using a multi-dimensional spectral clustering scheme over both the features and the supervised output spaces, and is suitable for classification as well as regression tasks. The FL aggregation operations support system (OSS) server then uses the reported dataset entropies to devise 1) an entropy-based federated averaging scheme, and 2) a stochastic participant selection policy to significantly stabilize the training, minimize the convergence time, and reduce the corresponding computation cost. Numerical results are provided to show the superiority of these novel approaches.

Keywords: dataset entropy, fast federated learning, non-iid, spectral clustering, stochastic policy, 6G

OPEN ACCESS

Edited by:
Mingzhe Chen,
Princeton University, United States

Reviewed by:
Zhaohui Yang,
King's College London,
United Kingdom
Sihua Wang,
Beijing University of Posts and Telecommunications (BUPT), China

***Correspondence:**
Brahim Aamer
aamer.brahim@gmail.com

Specialty section:
This article was submitted to Data Science for Communications, a section of the journal Frontiers in Communications and Networks

Received: *10 July 2021*
Accepted: *25 August 2021*
Published: *07 October 2021*

Citation:
Aamer B, Chergui H, Benjillali M and Verikoukis C (2021) Entropy-Driven Stochastic Federated Learning in Non-IID 6G Edge-RAN. Front. Comms. Net 2:739414. doi: 10.3389/frcmn.2021.739414

1 INTRODUCTION

6G wireless networks announces the era of massive heterogeneous digital services, that extend the vertical use cases to the final consumer, which is challenging from a network management point of view. Indeed, in this new context, classical centralized monitoring, analysis, and control would become impractical, as they usually represent a single point of failure and suffer from large overhead. Alternatively, decentralized service processing would bring scalability, low raw data exchange and therefore more system sustainability. In this regard, distributed artificial intelligence (AI) approaches, and in particular FL schemes, can play a pivotal role in leveraging the potential of scattered monitoring data across the network as well as the computing power of edge cloud, while reducing the computational costs and enabling fast local analysis and decision. Nevertheless, FL performance is often limited by the convergence delay due to several conceptual and operational issues that are reviewed in the sequel.

1.1 Related Work

In (Brendan McMahan et al., 2017), the authors have proposed the federated averaging (FedAvg) algorithm that synchronously aggregates the parameters, and is thus susceptible to the so-called

straggler effect, i.e., each training round only progresses as fast as the slowest edge device since the FL server waits for all devices to complete local training before the global aggregation can be performed. Alternatively, the asynchronous model in (Sprague et al., 2018) has been introduced to improve the scalability and efficiency of FL. For asynchronous FL, the server updates the global model whenever it receives a local update which grants more robustness against participants joining halfway during a training round, as well as when the federation involves participating devices with heterogeneous processing capabilities. However, the model convergence is found to be significantly delayed when data is non independent and identically distributed (non-IID) and unbalanced (Zhao et al., 2018). To solve this issue, it has been proposed to distribute a public dataset to the FL clients at the beginning. However, such a dataset may not always exist, or the participants may refuse to download them for security reasons. Therefore, an alternative solution was to construct an approximately IID dataset using inputs from a limited number of privacy insensitive participants (Yoshida et al., 2019). In the Hybrid-FL protocol, the server asks random participants if they allow their data to be uploaded. During the participant selection phase, apart from selecting participants based on computing capabilities, participants are selected such that their uploaded data can form an approximately IID dataset in the server, i.e., the amount of collected data in each class has close values. Thereafter, the server trains a model on the collected IID dataset, and merges this model with the global model trained by the participants. Nevertheless, requests for data sharing are not in line with the original intent of FL. As an improvement, the authors in (Xie et al., 2019) have proposed the FedAsync algorithm in which newly received local updates are adaptively weighted according to staleness, that is defined as the difference between the current epoch and the iteration to which the received update belongs to. For example, a stale update from a straggler is outdated since it should have been received in previous training rounds. As such, it is given a smaller weight. In addition, the authors prove the convergence guarantee for a restricted family of non-convex problems. However, the current hyperparameters of the FedAsync algorithm still have to be tuned to ensure convergence in different settings. Hence, the algorithm is still unable to generalize to suit the dynamic computation constraints of heterogeneous devices. Given this uncertainty surrounding the reliability of asynchronous FL, synchronous FL remains the most commonly used approach (Keith et al., 2019). In this context, it has been confirmed that the correlation between the model parameters of different clients is increasing as the training progresses, which implies that aggregating parameters directly by averaging may not be a reasonable approach in general (Xiao et al., 2020). Besides, a fair resource federated learning approach has been studied recently in (Tian et al., 2020), which introduces a weighted averaging that gives higher weights to devices with the worst performance (i.e., the largest loss) to let them dominate the objective, and thereby impose more uniformity to the training accuracy. Finally, authors in (Niknam et al., 2019) and (Yang et al., 2021) have listed the different FL motivations, challenges and applications on 6G and wireless communications, where FL has been presented as a solution to address energy, bandwidth, delay and privacy questions in wireless communications. As energy consumption is one of the important aspects to consider in FL, in (Tran et al., 2019) the trade-off between learning time, learning accuracy and terminals power consumption has been investigated.

1.2 Contributions

In this paper, our contribution is two-fold.

- We first introduce the concept of the *entropy* of a dataset in both classification and regression tasks, where we jointly consider the features and supervised outputs to characterize the distribution of its samples and the underlying quantity of information based on a custom spectral clustering strategy. This generalized entropy captures the diversity of a dataset as well as its unbalanced structure and non-IIDness.
- By leveraging the proposed entropy as an *a priori* information, we develop two novel FL strategies to make central units (CUs) at 6G Edge-RAN collaborate in learning a certain resource usage, namely, 1) Entropy-weighted federated aggregation which involves all the CUs in the FL training task while prioritizing the most balanced and uncorrelated datasets (i.e., those maximizing the entropy) and 2) Entropy-driven stochastic policy for selecting only a subset of CUs to take part in the FL task. This consists on sampling, at each FL round, the active CUs according to an entropy-based probability distribution, which dramatically reduces the convergence stability and time, as well as the underlying resource consumption by avoiding concurrent training by all CUs at each round.

2 NETWORK DESCRIPTION AND DATA COLLECTION

2.1 Edge-RAN

As depicted in **Figure 1**, the considered network corresponds to a 6G edge-RAN under the central unit (CU)/distributed unit (DU) functional split, where each transmission/reception point (TRP) is co-located with its DU, while all CUs are hosted in an edge cloud where they run as virtual network functions (VNFs). Each CU k ($k = 1, \ldots, K$) performs RAN key performance indicators (KPIs) data collection to build its local dataset $\mathcal{D}_k = \{\mathbf{x}_k^{(i)}, y_k^{(i)}\}_{i=1}^{D_k}$ of size D_k, where $\mathbf{x}_k^{(i)}$ stands for the input features vector while $y_k^{(i)}$ represents the corresponding output. Given that this dataset is generally non-exhaustive to train accurate analytical models, the CU takes part in a federated learning task wherein an OSS server—located at the core cloud—plays the role of a model aggregator. In this work, the CUs and the OSS are connected via fiber transport links, which present a very stable behavior (compared to the wireless channel), and have no effect on the accuracy of the FL.

2.2 Data Collection

Table 1 shows the features and the supervised output of the local datasets, which have been collected from a live LTE-Advanced (LTE-A) RAN with a granularity of 1 h. The considered TRPs cover areas

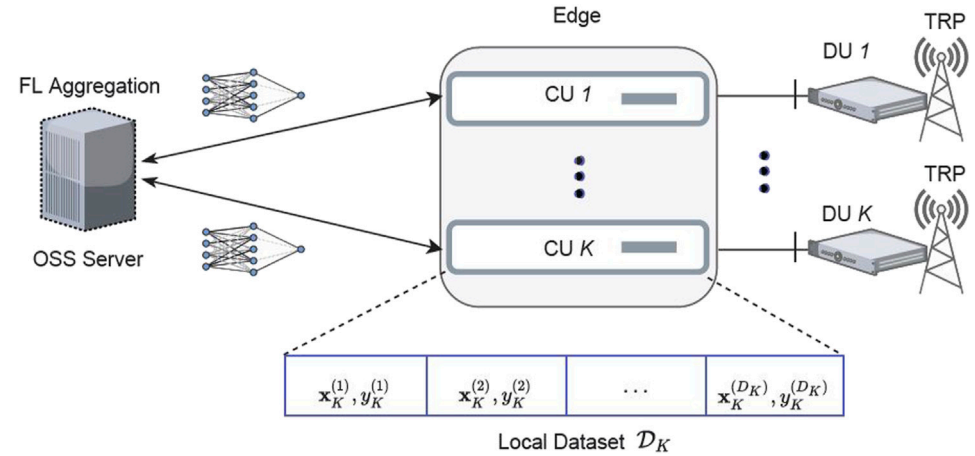

FIGURE 1 | Network architecture.

TABLE 1 | Dataset features and output.

Feature	Description
Cell Throughput	Cell Downlink Average Throughput
User Throughput	User Downlink Average Throughput
BLER	Average Block Error Rate
# Users	Downlink Average Active Users
MIMO Rank	Average MIMO Rank
DL PRB	Downlink PRB Usage Percentage
TA	Average Timing Advance
CQI	Average Channel Quality Index
QPSK	Percentage of QPSK modulation usage
Output	**Description**
Traffic	Traffic Volume (Output)

with different traffic profiles—both in space and time—that tightly depend on the heterogeneous users distribution and behavior in each context (e.g., residential zones, business zones, entertainment events, . . .). On the other hand, the radio KPIs are correlated with the time-varying channel conditions. These realistic datasets are therefore non-IID, which is more challenging for FL algorithms as studied in (Li et al., 2021).

3 PROPOSED ENTROPY-BASED FEDERATED LEARNING

To tackle the FL convergence in practical non-IID setups, we seek an objective and compressed metric capturing both the distribution of a dataset and its quantity of information, while not depending on the local models. In this regard, we introduce the notion of *dataset entropy* that is a sufficient statistic to characterize the unbalanced structure of a dataset, as well as its independence from other datasets. Specifically, the entropy is maximized under a uniform distribution with low probability mass function (PMF). By relying on the *a priori* entropies of all CUs, the aggregation server can implement novel CUs selection and models combining schemes to accelerate and stabilize the FL convergence.

3.1 Dataset Entropy

Since we are targeting a generalized definition of the entropy, the labels of a classification dataset are not reliable to reflect the distribution of data since it does not apply to regression tasks where the supervised output is continuous, and it omits the effect of the input features. In particular, samples with different feature values but presenting approximately similar outputs are not providing the same information and might not necessarily fit in the same group of data. Therefore, in order to accurately discern the samples, we consider a joint approach where both the features and the supervised output are used. To that end, each CU uses a clustering algorithm that operates on the so-called *similarity matrix* $\mathbf{S}_k$ whose entries measure the logical correlations between the dataset samples vectors including both the features and the supervised output, i.e.,

$$\tilde{\mathbf{x}}_k^{(i)} = \left[\mathbf{x}_k^{(i)}\, y_k^{(i)}\right]. \tag{1}$$

This matrix is built using a radial basis function (RBF) kernel with parameter σ. As such, the (i, j)-th matrix element is given by

$$s_{i,j}^{(k)} = \exp\left(-\frac{d\left(\tilde{\mathbf{x}}_k^{(i)}, \tilde{\mathbf{x}}_k^{(j)}\right)}{\sigma^2}\right), \tag{2}$$

where d stands for the pairwise logical distance between samples' vectors i and j. Let F denote the number of features in the datasets. A general definition of this distance that involves both the features and the output can be written as

$$d\left(\tilde{\mathbf{x}}_k^{(i)}, \tilde{\mathbf{x}}_k^{(j)}\right) = \sum_{f=1}^{F+1} \alpha_f |x_k^{(i,f)} - x_k^{(j,f)}|, \tag{3}$$

where $\{\alpha_f\}$ stand for the weights of each feature/output in the distance and verify $\sum_{f=1}^{F+1} \alpha_f = 1$. They can be fine-tuned to orient the clustering towards the direction of the most relevant features or prioritize the output. For the sake of simplicity, and to avoid

generating a high number of scenarios, we settle in this work to the typical setting where the weights are uniform, i.e., $\alpha_f = 1/(F+1)$. Further investigation on the effect of the weights on the entropy is left for future works.

Since basic clustering algorithms usually require the target number of clusters as an input, we resort to the well-established self-tuning spectral clustering (STSC) technique that presents a time complexity of $\mathcal{O}(D_k^3)$, but is still practical as long as the dataset size $D_k < 10^3$ (Tsironis et al., 2013). Since in our case we have only small datasets for each CU (e.g., of size 100)—which is by the way one of the reasons to resort to federating learning, the clustering scheme is viable in our case.

The STSC relies on the eigenvalues and eigenvectors of the similarity matrix. To that end, we first define $\mathbf{\Lambda}$ to be a diagonal matrix with

$$\mathbf{\Lambda}_{i,i}^{(k)} = \sum_{j=1}^{D_k} s_{i,j}^{(k)}, \tag{4}$$

and construct the normalized affinity matrix

$$\mathbf{L}_k = \mathbf{\Lambda}^{-1/2}\mathbf{S}_k\mathbf{\Lambda}^{-1/2}. \tag{5}$$

When $\mathbf{\Lambda}$ is strictly block diagonal, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks padded appropriately with zeros. Let $\mathbf{X}_k$ denotes the block diagonal matrix gathering the eigenvectors. In this case, we can automatically cluster a dataset into an appropriate number of clusters that minimizes a custom cost function defined in terms of the coefficients of a rotated and normalized version of matrix $\mathbf{X}_k$ (Zelnik and Pietro, 2004). Let us assume that for CU k, the clustering yields n_k clusters $\mathcal{C}_{k,1}, \ldots, \mathcal{C}_{k,n_k}$ with probabilities $\Pr(\mathcal{C}_{k,1}), \ldots, \Pr(\mathcal{C}_{k,n_k})$ over dataset $\mathcal{D}_k$, which are calculated via the number of samples per cluster $\Delta_{k,p}$ as

$$\Pr\left(\mathcal{C}_{k,p}\right) = \frac{\Delta_{k,p}}{D_k}. \tag{6}$$

The Corresponding Entropy Is Then Defined as

$$\varepsilon_k = -\sum_{p=1}^{n_k} \Pr\left(\mathcal{C}_{k,p}\right)\log\left\{\Pr\left(\mathcal{C}_{k,p}\right)\right\}. \tag{7}$$

By letting the CUs report their dataset entropies $\{\varepsilon_k\}_{k=1}^K$ to the aggregation server before starting the training, it becomes possible to devise advanced entropy-driven FL strategies that prioritize the CUs with high entropy datasets.

3.2 Entropy-Driven FL Combining

In this strategy, the aggregation server directly uses the entropies to perform a weighted averaging of all CUs local models at each round t, i.e.,

$$\mathbf{W}^{(t+1)} = \sum_{k=1}^{K} \frac{\varepsilon_k}{\bar{\varepsilon}}\mathbf{W}_k^{(t)}, \tag{8}$$

where

$$\bar{\varepsilon} = \sum_{p=1}^{K} \varepsilon_p \tag{9}$$

is the cumulative sum of the different CUs entropies that serves as a factor. This allows the CUs with high entropies to dominate and orient the FL training, although this requires the participation of all CUs.

Algorithm 1: Entropy-Driven Stochastic Federated Learning Policy.

Input: K, m, η, T, L. # See Table II
parallel for $k = 1, \ldots, K$ **do**
Calculate RBF-based Similarity Matrix entries
$s_{i,j}^{(k)} = \exp\left(-\frac{d\left(\tilde{\mathbf{x}}_k^{(i)}, \tilde{\mathbf{x}}_k^{(j)}\right)}{\sigma^2}\right)$, $1 \le i, j \le D_k$
Construct the normalized affinity matrix
$\mathbf{\Lambda}_{i,i}^{(k)} = \sum_{j=1}^{D_k} s_{i,j}^{(k)}$
$\mathbf{L}_k = \mathbf{\Lambda}^{-1/2}\mathbf{S}_k\mathbf{\Lambda}^{-1/2}$
Form the block diagonal $\mathbf{X}_k$ of the eigenvectors of $\mathbf{L}_k$
Clustering
CU k clusters $\mathcal{D}_k$ based on matrix $\mathbf{X}_k$
return n_k, $\Pr\left(\mathcal{C}_{k,1}\right), \ldots, \Pr\left(\mathcal{C}_{k,n_k}\right)$
Calculate Dataset Entropy
$\varepsilon_k = -\sum_{p=1}^{n_k} \Pr\left(\mathcal{C}_{k,p}\right)\log\Pr\left(\mathcal{C}_{k,p}\right)$
CU k reports ε_k to the aggregation server
end parallel for
# Federated Learning
# Server Generates Probability Distribution
for $k = 1, \ldots, K$ **do**
  $\pi_k = \frac{\exp\{\varepsilon_k\}}{\sum_{l=1}^{K}\exp\{\varepsilon_l\}}$, $k = 1, \ldots, K$
end
Server initializes $\mathbf{W}^{(0)}$ with random Gaussian weights
for $t = 0, \ldots, T-1$ **do**
  # Server Samples the m CUs
  $\mathrm{CU}_{k_1}^{(t)}, \ldots, \mathrm{CU}_{k_m}^{(t)} \sim \{\pi_1, \ldots, \pi_K \mid \mathrm{CU}_1, \ldots, \mathrm{CU}_K\}$
  Server broadcasts $\mathbf{W}^{(0)}$ to the m selected CUs
  parallel for $k \in \{k_1, \ldots, k_m\}$ **do**
  # Local epochs
  for $l = 0, \ldots, L-1$ **do**
    $\mathbf{W}_{k,l} = \mathbf{W}_{k,l-1} - \eta\nabla\mathcal{L}\left(\mathbf{W}, \mathbf{b}\right)$
  end
  return $\mathbf{W}_k^{(t)} = \mathbf{W}_{k,L-1}$
  Each local CU k sends $\mathbf{W}_k^{(t)}$ to the aggregation server.
  end parallel for
  # Server Aggregation
  return $\mathbf{W}^{(t+1)} = \sum_{k\in\{k_1,\ldots,k_m\}} \frac{D_k}{D}\mathbf{W}_k^{(t)}$
  Broadcasts $\mathbf{W}^{(t+1)}$ to all K CUs.
end

3.3 Entropy-Driven Stochastic FL Policy

To optimize the federated learning computation time as well as the underlying resource consumption, we aim at selecting only a number of active CUs in each FL round. In this respect, we introduce an entropy-driven stochastic CU selection policy wherein the aggregation server first generates a probability distribution over all the CUs using their received entropies. In fact, CUs with high entropies hold datasets that are rich in terms of quantity of information and can lead to more generalized models in the training. A direct strategy would consist on selecting the m CUs with highest entropies during all the training. But since the datasets of CUs with low entropy can also hold samples that are non-existing in the other high entropy datasets and yet can help in further generalizing the FL model, the idea we have proposed is to give them a chance by implementing a softmax stochastic policy, where each CU can participate in the training with a probability proportional to its entropy. Hence, in the long-term, even CUs with low entropy are given a chance in

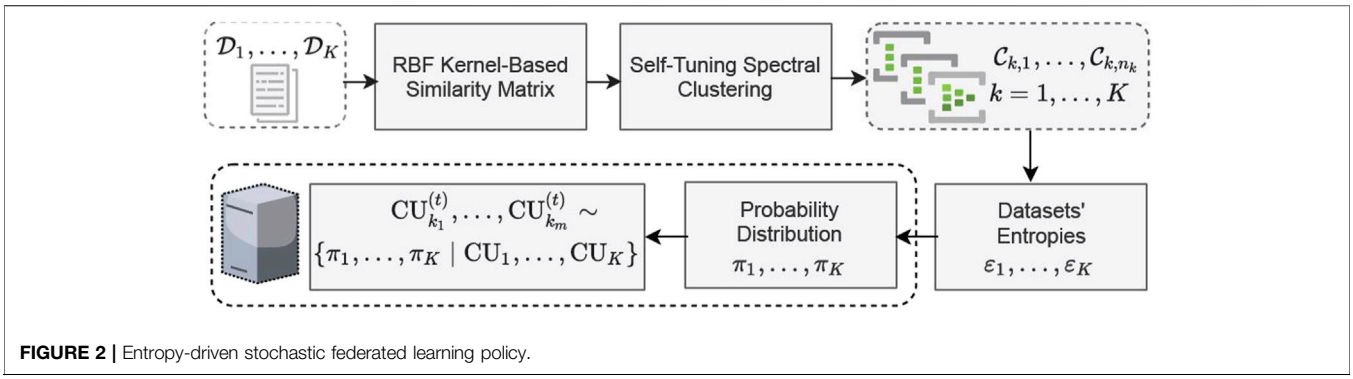

FIGURE 2 | Entropy-driven stochastic federated learning policy.

TABLE 2 | FL settings.

Parameter	Description	Value
T	Number of rounds	20
L	Number of epochs	50
K	Number of CUs	6
M	Number of selected CUs	3
D_k	Local dataset size	100
η	Learning rate	0.001
Σ	Kernel parameter	1.0

some rounds to train the model. This leads to a fast convergence (since it orients the training to the CUs with high potential) while ensuring a more general model at the end.

This is achieved by a direct softmax activation layer, i.e.,

$$\pi_k = \frac{\exp\{\varepsilon_k\}}{\sum_{p=1}^{K} \exp\{\varepsilon_p\}}. \tag{10}$$

Next, at each FL round t, as illustrated in **Figure 2**, the server selects a subset of $m < K$ CUs to participate in the training by sampling the non-uniform CUs set with probabilities $\{\pi_1, \ldots, \pi_K\}$, i.e.,

$$\mathrm{CU}_{k_1}^{(t)}, \ldots, \mathrm{CU}_{k_m}^{(t)} \sim \{\pi_1, \ldots, \pi_K \mid \mathrm{CU}_1, \ldots, \mathrm{CU}_K\}, \tag{11}$$

which ensures that, by the convergence round, the CUs would have stochastically taken part in the FL task according to the initial probability distribution, while avoiding the concurrent training by all CUs at each round. In this case, the model averaging at round t is performed as

$$\mathbf{W}^{(t+1)} = \sum_{k \in \{k_1, \ldots, k_m\}} \frac{D_k}{D} \mathbf{W}_k^{(t)}. \tag{12}$$

Where D is the total samples over all CUs datasets. This entropy-driven stochastic policy is summarized in **Algorithm 1**, where $\mathcal{L}(\cdot, \cdot)$ stands for the mean square error (MSE) loss function, and **b** is the bias, while the rest of FL setting parameters is provided in **Table 2**.

4 STOCHASTIC FEDERATED LEARNING CONVERGENCE ANALYSIS

In this section, we analyze the convergence probability of the stochastic federated learning. In this intent, a closed-form expression for the lower bound of the convergence probability is derived, reflecting the effects of the CUs selection probability and the datasets sizes.

Theorem 1 (Convergence Analysis of the Stochastic Federated Learning). *Consider that the CUs selection in the stochastic federated learning follows a policy* $\{\pi_1, \ldots, \pi_K\}$, *and let* Ω *and* B_k *stand for the upper bounds on the weights and the* norm *of subgradient* $\nabla \mathcal{L}(\mathbf{W}_k^{(t)})$, *respectively. Let* $\alpha_k \sim \mathcal{B}(\pi_k)$ *denote the CU activation bit. Then, the federated learning convergence probability satisfies*

$$\mathbf{Pr}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left(\mathcal{L}\left(\mathbf{W}^{(t)}\right) - \mathcal{L}\left(\mathbf{W}^{*}\right)\right) < \epsilon\right] \geq \phi(\epsilon), \tag{13}$$

where

$$\phi(\epsilon) = 1 - \exp - \left\{\frac{T\epsilon^2}{2\left(2\sum_{k=1}^{K} \pi_k \frac{D_k}{D} B_k \Omega\right)^{2^*}}\right\} \tag{14}$$

Proof. First, by means of the subgradient inequality we have at round t:

$$\mathcal{E}^{(t)} = \mathcal{L}\left(\mathbf{W}^{(t)}\right) - \mathcal{L}\left(\mathbf{W}^{*}\right) \leq \langle \nabla \mathcal{L}, \mathbf{W}^{*} - \mathbf{W}^{(t)} \rangle. \tag{15}$$

Using Cauchy-Schwarz inequality, we get

$$\mathcal{E}^{(t)} \leq \left\|\nabla \mathcal{L}\left(\mathbf{W}^{(t)}\right)\right\| \left\|\mathbf{W}^{(t)} - \mathbf{W}^{\star}\right\|. \tag{16}$$

By recalling the federated learning aggregation **Eq. 12**, we can write

$$\nabla \mathcal{L}\left(\mathbf{W}^{(t)}\right) = \sum_{k=1}^{K} \alpha_k \frac{D_k}{D} \nabla \mathcal{L}\left(\mathbf{W}_k^{(t)}\right). \tag{17}$$

Therefore, from **Eqs 16**, **17** and by invoking the triangle inequality we have

$$\begin{aligned} \mathcal{E}^{(t)} &\leq \sum_{k=1}^{K} \alpha_k \frac{D_{k,n}}{D_n} \left\|\nabla \mathcal{L}\left(\mathbf{W}_k^{(t)}\right)\right\| \left\|\mathbf{W}^{(t)} - \mathbf{W}^{\star}\right\| \\ &\leq 2 \sum_{k=1}^{K} \alpha_k \frac{D_k}{D} B_k \Omega \end{aligned} \tag{18}$$

By the monotonicity of the expectation, we have

$$\mathbf{E}(\mathcal{E}^{(t)}) \leq 2 \sum_{k=1}^{K} \pi_k \frac{D_k}{D} B_k \Omega = C. \tag{19}$$

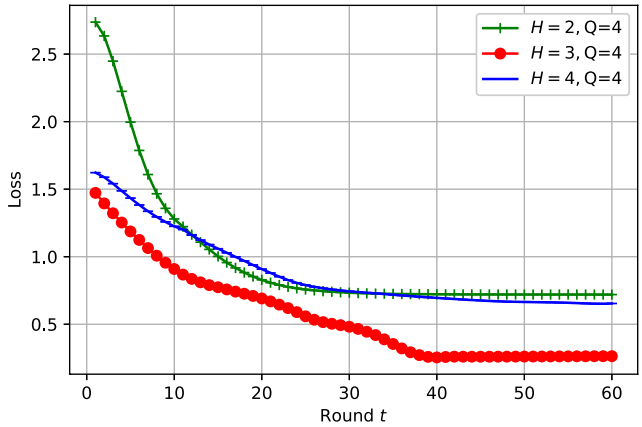

FIGURE 3 | Different number of layers configurations comparison for η = 0.001.

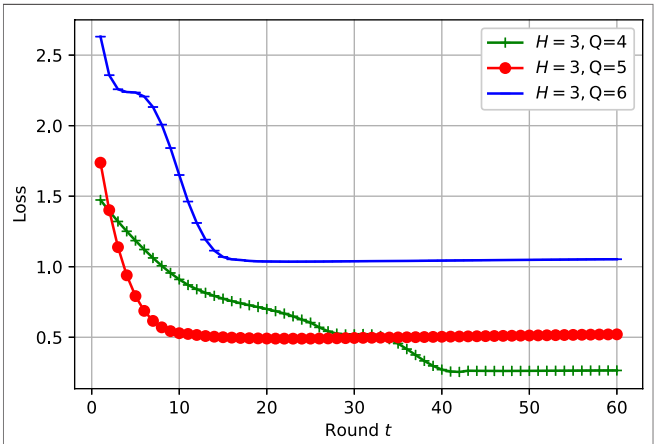

FIGURE 4 | Different number of neurons per layer configurations comparison for η = 0.001.

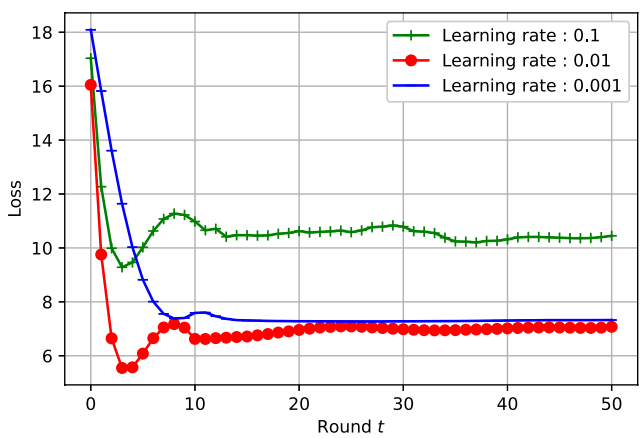

FIGURE 5 | Convergence of entropy-weighted vs. learning rate.

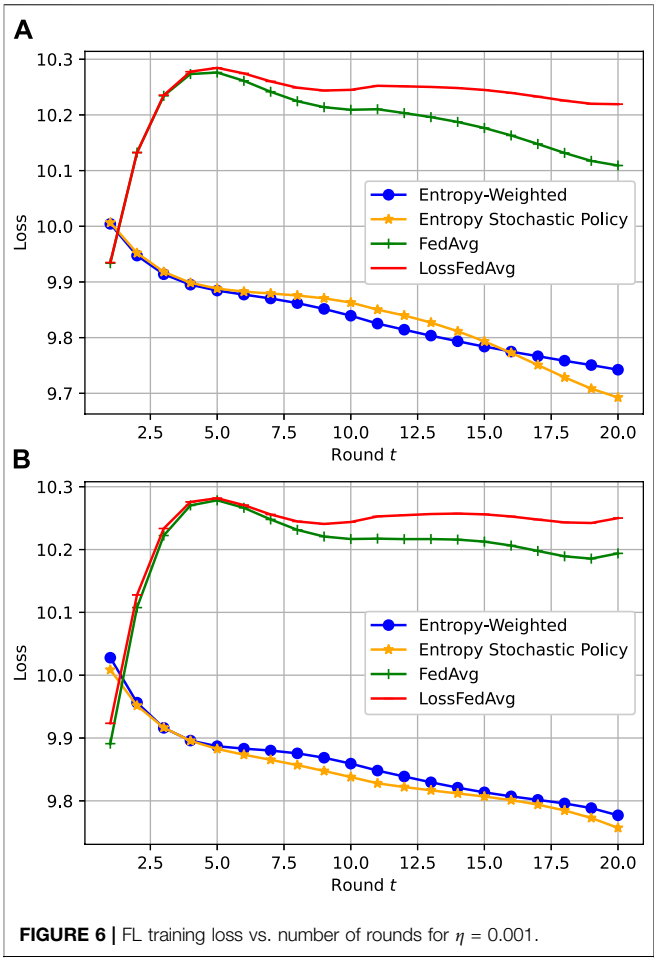

FIGURE 6 | FL training loss vs. number of rounds for η = 0.001.

By means of Hoeffding-Azuma's inequality (Hoeffding, 1963), we have

$$\Pr\left[\frac{1}{\tau}\sum_{t=1}^{\tau}\mathbf{E}(\mathcal{E}^{(t)}) < \epsilon \mid \tau = T\right] \geq 1 - \exp - \left\{\frac{T\epsilon^2}{2C^2}\right\}, \tag{20}$$

5 NUMERICAL RESULTS

5.1 Settings and Baselines

5.1.1 DNN Setting

The structure of the global model weights matrix $\mathbf{W}$ has been defined by the server to satisfy the findings of (Ke and Liu, 2008), where the authors have estimated the required number Q of neurons per layer based on the number H of hidden layers, the dataset sizes D_k, and the number of features F as

$$Q = \frac{F + \sqrt{\max_{k=1,\ldots,K} D_k}}{H}, \tag{21}$$

which is confirmed via **Figures 3**, **4**, where the best setting of the DNN model neurons turns out to be $Q = 4$ for $H = 3$. As a benckmark, the performance of our proposed approaches is compared with LossFedAvg (Li et al., 2021) and FedAvg (Brendan McMahan et al., 2017). FL settings are listed on

TABLE 3 | Results: Datasets clustering.

CU number	Balanced			Unbalanced		
	Nb samples	Nb clusters	Entropy	Nb samples	Nb clusters	Entropy
1	100	2	0.692	100	2	0.692
2	100	2	0.592	70	3	1.026
3	100	2	0.676	90	2	0.515
4	100	3	0.998	80	4	1.238
5	100	3	1.051	50	3	1.068
6	100	2	0.676	60	2	0.690

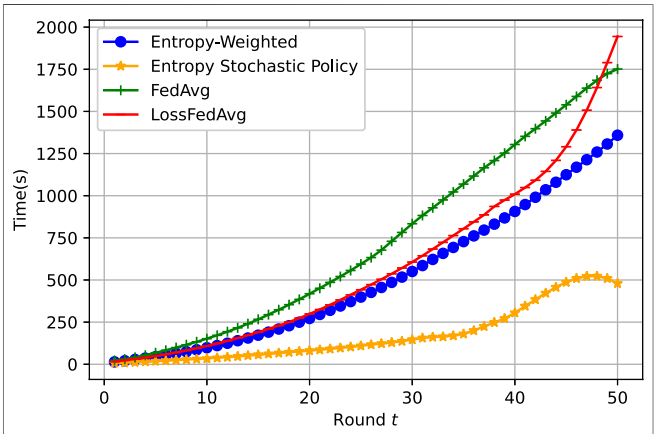

FIGURE 7 | Models time convergence for $\eta = 0.001$.

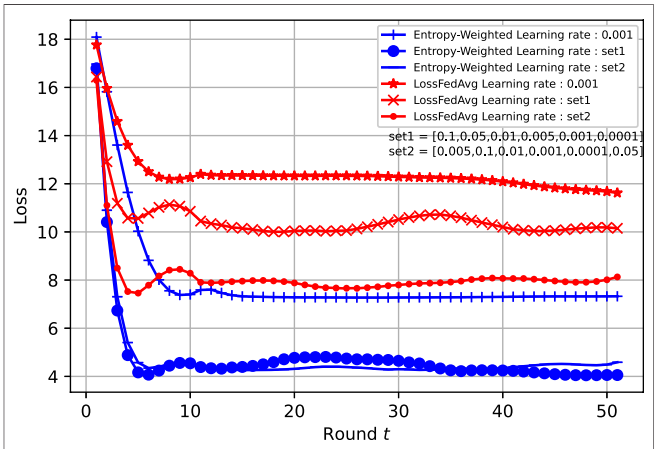

FIGURE 8 | Models convergence for heterogeneous learning rate sets.

Table 2, where FL system consists of $K = 6$ DUs running local DNN with a learning rate $\eta = 0.001$ for $T = 20$ rounds.

5.1.2 Learning Rate

The learning rate is a key parameter in ML models, therefore we have to select carefully its right value. In this perspective, we have simulated different learning rate values to illustrate Entropy-Weighted model convergence behaviour. In this respect, **Figure 5** shows fast convergence of Entropy-Weighted model with learning rate $\eta = 0.01$, while for $\eta = 0.001$ it is showing a stable yet more slow convergence to the same loss as the case of $\eta = 0.01$. Note that the adopted DNN optimizer is *Adam optimizer* (Kingma and Ba, 2015).

5.2 Numerical Results Analysis

5.2.1 Convergence

Figures 6A,B illustrate the gains achieved by the entropy-weighted approach compared to the baseline FedAvg and LossFedAvg. The comparison is done for both balanced and unbalanced non IID datasets. As showcased in **Table 3**, the entropy metric varies in balanced datasets, since the clustering technique takes into account the correlation between features as well as the supervised output. In the unbalanced scenario, the entropy difference between CUs is even clearer and demonstrates also that datasets with smaller size can sometimes yield more clusters compared to larger datasets, which further corroborates the role of the introduced entropy metric in characterizing a dataset efficiently.

A slightly lower losses are met with the entropy-weighted approach rather than the entropy stochastic policy, but both methods have the same convergence trend. In **Figure 6A,B** both entropy-based FL converge faster than FedAvg and LossFedAvg. Knowing how critical is the bandwidth occupation for FL exchanges, and how the CUs local model training is power consuming, especially in 6G mobile systems, our introduced entropy stochastic policy shows good results. This aspect becomes more critical if the FL result is an input for fast decision-making algorithms such as network slicing orchestration or resources scheduling.

Better than FedAvg and LossFedAvg, the entropy stochastic policy convergence trend is oscillating around entropy-weighted as in **Figure 6A,B**.

5.2.2 Time Complexity and Scalability

Another important achievement with the entropy stochastic policy is the reduction of the required time for a given number of rounds and exchanges between the OSS server and the CUs towards convergence, as shown in **Figure 7**, wherein the convergence time difference between the entropy-weighted approach and the entropy stochastic policy is exponentially growing with the number of FL rounds. Note that the corresponding wall-clock time performance is tightly dependent on the computation capabilities of both the OSS server and the CUs, but it shows that the stochastic policy FL minimizes the computation burden by selecting only a subset of CUs to take part in the training according to their *prior* entropy

measure, no matter how the number of CUs grows in the network. This proves the scalability of the proposed stochastic FL in large-scale deployments scenarios. More results can be generated for different values of K and m.

5.2.3 Learning Rate Sets

We have trained both entropy-weighted and LossFedAvg models using specific learning rate per each FL CU. As illustrated in **Figure 8**, better convergence is achieved with both used sets of learning rates compared to fixed $\eta = 0.001$. Where *set1* is a random selection of CUs learning rates, while in *set2*, η has been chosen according to each CU's entropy value, i.e., CUs with high entropy are assigned small η values and vice-versa. Note that the random learning rate strategy exhibits unstable convergence since it allows CUs with low entropy to learn faster and therefore dominate in some cases.

6 CONCLUSION

In this paper, we have introduced a novel *a priori* metric termed *dataset entropy* to characterize the distribution, the quantity of information, the unbalanced structure and the "non-IIDness" of a dataset independently of the models. This entropy is calculated via a generalized clustering strategy that relies on a custom similarity matrix defined over both the features and the supervised output spaces, and supporting both classification and regression tasks. The entropy metric has been then adopted to develop 1) an entropy-based federated averaging scheme, and 2) a stochastic CU selection policy to significantly stabilize the training, minimize the convergence time, and reduce the corresponding computation cost. Numerical results have been provided to corroborate these findings. In particular, the convergence time difference between Entropy-Weighted and Entropy Stochastic Policy schemes is exponentially growing with the number of FL rounds. Another important result is Entropy Stochastic Policy model convergence, which is better than FedAvg and LossFedAvg and oscillating near Entropy-Weighted model.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not readily available because the dataset is protected by IPR of the operator. Requests to access the datasets should be directed to aamer.brahim@gmail.com.

AUTHOR CONTRIBUTIONS

BA: Algorithms proposal and implementation. HC: Stochastic policy proposal. MB: Results analysis and recommendations proposal. CV: Results analysis and recommendations proposal.

REFERENCES

Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). "Communication-efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR: W&CP 54.

Hoeffding, A. (1963). Probability Inequalities for Sums of Bounded Random Variables. *J. Am. Stat. Assoc.* 58 (301), 13–30.

Ke, J., and Liu, X. (2008). "Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction," in IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China, December 19-20, 2008. doi:10.1109/paciia.2008.363

Keith, B., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., et al. (2019). Towards Federated Learning at Scale: System Design. [Online]. Available at: arxiv.org/abs/1902.01046.

Kingma, D. P., and Ba, J. (2015). "Adam: A Method for Stochastic Optimization," in 3rd International Conference for Learning Representations, San Diego, Jul. 2015.

Li, Q., Diao, Y., Chen, Q., and He, B. (2021). Federated Learning on Non-IID Data Silos: An Experimental Study. *Comput. Sci.*

Niknam, S., Dhillon, H. S., and Reed, J. H. (2019). Federated Learning for Wireless Communications: Motivation, Opportunities and Challenges. *Electr. Eng. Syst. Sci.* 58 (6), June 2020 46–51. doi:10.1109/MCOM.001.1900461

Sprague, M. R., Jalalirad, A., Scavuzzo, M., Capota, C., Neun, M., Do, L., and Kopp, M. (2018). "Asynchronous Federated Learning for Geospatial Applications," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, March, 2019 (Springer) 967, 21–28. doi:10.1007/978-3-030-14880-5_2

Tian, Li., Sanjabi, M., Ahmad, B., and Smith, V. (2020). *Fair Resource Allocation in Federated Learning*. Addis Ababa, Ethiopia: ICLR.

Tran, N. H., Bao, W., Albert, Z., Minh, N., Hong, C. S., and Nguyen, H. (2019). "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," in IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, April 29–May 2, 2019.

Tsironis, S., Sozio, M., Vazirgiannis, M., and Poltechnique, L. (2013). "Accurate Spectral Clustering for Community Detection in Mapreduce," in ÂŽÂŽ NIPS Workshops, Serbia, September 2018. doi:10.1109/SISY.2018.8524662

Xiao, P., Cheng, S., Stankovic, V., and Vukobratovic, D. (2020). Averaging Is Probably Not the Optimum Way of Aggregating Parameters in Federated Learning. *Entropy (Basel)* 22 (3). doi:10.3390/e22030314

Xie, C., Koyejo, S., and Gupta, I. (2019). Asynchronous Federated Optimization. [Online]. Available at: arxiv.org/abs/1903.03934.

Yang, Z., Chen, M., Kai-Kit Wong, H., Poor, V., and Cui, S. (2021). Federated Learning for 6G: Applications, Challenges, and Opportunities. *Comput. Sci.*

Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., and Yonetani, R. (2019). Hybrid-FL: Cooperative Learning Mechanism Using Non-IID Data in Wireless Networks. [Online]. Available at: arxiv.org/abs/1905.07210.

Zelnik, L., and Pietro, M. (2004). "Self-Tuning Spectral Clustering," in the 17th International Conference on Neural Information Processing Systems (NIPS'04), Vancouver, January 2004, 1601–1608.

Zhao, Y., Meng, L., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated Learning with Non-IID Data. [Online]. Available at: arxiv.org/abs/1806.00582.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Aamer, Chergui, Benjillali and Verikoukis. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.