EntropicFL: Efficient Federated Learning via Data Entropy and Model Divergence

Rómulo W. C. Bustincio, Allan M. de Souza, Joahannes B. D. da Costa, Luiz F. Bittencout
Universidade Estadual de Campinas, Campinas, Brazil
r204185@dac.unicamp.br, {allanms, joahannes.costa, bit}@ic.unicamp.br

ABSTRACT

Federated Learning (FL) is a strategy for training distributed learning models. This approach gives rise to significant challenges including the non-independent and identically distributed (non-IID.) characteristics inherent to the training data, which can wield influence over the comprehensive accuracy of the global model. Moreover, the collaborative involvement of multiple clients in training protocols frequently engenders increased communication overheads and resource management overheads. In this research, we present an innovative strategy to address the inherent challenges of federated learning, including the communication overhead, data heterogeneity, and privacy preservation concerns. Our proposed approach centers on the concept of adaptive client selection, comprising a two-step process: firstly, the identification of a subset of clients possessing pertinent and representative data for participation in model training, and secondly, the determination of whether to transmit the local updates from these selected clients to the central aggregation server. Our methodology leverages the metrics of data entropy and model divergence to guide this client selection process. By applying this approach, we effectively mitigate communication overhead without compromising the accuracy achieved in the federated learning process.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; *Machine learning*; **Distributed artificial intelligence**.

KEYWORDS

Federated learning, Client selection, Heterogeneity, Entropy

ACM Reference Format:
Rómulo W. C. Bustincio, Allan M. de Souza, Joahannes B. D. da Costa, Luiz F. Bittencout. 2023. EntropicFL: Efficient Federated Learning via Data Entropy and Model Divergence. In *2023 IEEE/ACM 16th International Conference on Utility and Cloud Computing (UCC '23), December 4–7, 2023, Taormina (Messina), Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3603166.3632611

UCC '23, December 4–7, 2023, Taormina (Messina), Italy

ACM ISBN 979-8-4007-0234-1/23/12…$15.00
https://doi.org/10.1145/3603166.3632611

1 INTRODUCTION

With the ever-growing proliferation of edge devices, a myriad of opportunities emerge, and among them stands out Federated Learning (FL). In FL, multiple entities, referred to as clients, collaborate to address a machine learning problem. Federated Learning was first introduced by McMahan et al. [13], with its primary limitation being the associated communication costs. In FL, clients operate under the coordination of a central server or service provider, which may not be present in other distributed learning setups. Unlike traditional approaches, in FL, each client's data remain stored locally and is not transferred to the central server [11, 14]. Instead, only focused model updates, designed explicitly for immediate aggregation at the central server, are used to achieve the learning objective collectively. Communication plays a vital role within the Federated Learning ecosystem, as highlighted in previous studies [16], [17]. It is widely recognized that communication can emerge as a primary bottleneck in the context of FL, as emphasized in recent research [6]. Moreover, heterogeneity is recognized as a factor that can degrade the convergence of models [18]. In this sense, two distinct types of heterogeneity exist, namely, state heterogeneity and hardware heterogeneity, and both are related to the aforementioned problems [3]. The heterogeneity of state, which is not well-studied in the literature, is associated with factors such as CPU state (busy/free and dynamic), network connection stability, and other similar variables. These factors can introduce variations in the performance and behavior of individual devices participating in the FL. On the other hand, heterogeneity of hardware refers to differences in hardware characteristics among devices, such as CPU capabilities, RAM capacity, battery life, and other hardware-related aspects. These variations can affect the computational resources available for training and inference tasks, potentially impacting the overall performance and efficiency of the FL system [18].

Considering these factors, efforts are being made to enhance client selection by leveraging contextually derived information [9]. For instance, owing to the ever-changing attributes of present-day devices, there exists a limitation in the effectiveness of training quality. In conventional FL methods, a proactive approach to client scheduling is commonly employed. This involves the server selecting participants based on client state and associated parameters. Nevertheless, participants chosen through reactive methods are prone to higher chances of failure during an FL round [5].

FL is a collaborative framework where numerous clients work together to address machine learning problems, all overseen by a central aggregator (server) [19]. In the research conducted by Guendouzi *et al.* [4], the FL process can be broken down into several distinct phases. These phases encompass problem identification, participant selection, training, parameter sharing, parameter aggregation, and parameter broadcast. It is essential to emphasize that

each phase introduces specific and distinctive challenges. Client selection defines the set of clients that will train the model, and a well-designed client selection scheme in Federated Learning can substantially enhance model accuracy [8]. In this sense, the focuses of this study are: *(i)* client selection and *(ii)* parameter sharing. In this way, the communication overhead is a key challenge in FL [12]. Several research efforts have tackled this challenge using thresholds, client reduction, and multi-criteria approaches [1, 17]. However, it is worth noting that many of these methods require a comprehensive understanding of the data in order to select an appropriate configuration. In special, client selection defines the set of clients that will train the model, which is a crucial step in ensuring the effectiveness of the process when statistically heterogeneous data exist in the set of clients. The presence of non-IID data, characterized by variations in both data amount and category distribution, often leads to significant degradation in accuracy for many deep neural network models trained using distributed methods [10].

In this context, this work introduces EntropicFL, which uses a client selection strategy based on two criteria, namely the entropy of data and the accuracy of the clients. Given that entropy is directly related to non-IID nature of the on-devices training data [15], using the entropy criterion could be a promising approach to develop a client selection strategy. EntropicFL addresses the issue of non-IID data to ensure model convergence, reducing the total number of updates transferred of selected clients. Consequently, communication overhead and bottlenecks are reduced. In our experimental results, EntropicFL has demonstrated achieving accuracy very close to the methods in the literature, while reducing communication overhead by at least 27.7%.

This paper is organized as follows. Section 2 provides an overview of the relevant literature and theoretical foundations of the proposed algorithm. In Section 3, basic concepts are described. Section 4 provides a comprehensive overview of the design of the proposal. Section 5 describes the implementation and the configuration settings of the experiments that were conducted, and finally, Section 6 presents the conclusions and future works.

2 RELATED WORK

This Section describes a set of related work considering the client selection problem in FL environments. Wang *et al.* [17] introduced CMFL (Communication-Mitigated Federated Learning). CMFL employs sign-based metrics to determine whether to transmit local updates to the server, contingent on a threshold. This threshold is reliant on data distribution and necessitates an exploratory investigation. CMFL also guarantees convergence. In contrast, EntropicFL dynamically computes the threshold in each round using divergence weights between local and global models, allowing each client to determine whether to send updates to the server, consequently reducing the communication overhead.

In Zhao et. al [21], weight divergence in non-IID data is examined, revealing higher divergence compared to IID data. They propose enhancing accuracy in non-IID data by establishing a shared, small global dataset among clients. Our research leverages model divergence to detect irrelevant client updates.

The FedMCCS framework, introduced by Abdulrahman et al. in their 2021 work (Abdulrahman, 2021) [1], focuses primarily on improving client selection and participation in federated learning, particularly addressing issues related to client diversity and resource constraints. However, one significant aspect that has not been addressed in this framework is the optimization of client model update efficiency during the federated learning process. In our research, we introduce an approach where we integrate a divergence model to predict the influence of each client's update on the global model and decide if this contribution is relevant or irrelevant to consider communicating to the server.

CSFedAvg in [20] utilizes weight divergence to measure the non-iid degree and prefers participants with lower divergence. However, it necessitates server-side model training and auxiliary data, offering no communication overhead reduction assurance. In our proposal, clients determine server updates based on weight divergence criteria, and reducing communication overhead is guaranteed.

In the work by Orlandi [15], they introduce FedAvgBE a framework designed to mitigate the impact of non-IID by identifying problematic data blocks within clients' local datasets using mean global entropy calculations. While FedAvgBE effectively reduces communication costs, it falls short in addressing the issue of "communication overhead". This oversight could potentially necessitate an increase in the number of training rounds or the addition of more local epochs to improve the model's accuracy.

In the state of the art, the reduction of "communication overhead" is approached from multi-objective perspectives and through data analysis to establish server update criteria. However, it is important to note that data analysis and the search for appropriate thresholds may require additional experiments in some cases. This article presents EntropicFL, an approach designed to leverage entropy metrics for client selection, allowing us to gauge the data uncertainty associated with each client's dataset and select suitable participants for federated learning. Moreover, our methodology establishes criteria on the client side to determine whether an update should be communicated, effectively reducing communication overhead. This dual-pronged approach enhances the extraction of valuable insights from client data while safeguarding data privacy and optimizes the communication process.

3 BASIC CONCEPTS

In this section, we define the basic concepts, starting with a formal definition of FL and the concept of Shannon entropy.

3.1 Federated Learning Model

FL models usually necessitate an initial training process, typically encompassing the primary steps, as depicted in Figure 1. Consider, furthermore, that for each trained model, whether at the local or global level, there are associated weights. This arises from the fact that they are machine learning models.

3.2 Shannon Entropy

Shannon entropy plays a crucial role in quantifying uncertainty and characterizing the average information content of a data source, as discussed in [15]. The Shannon entropy is computed as follows.

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{1}$$

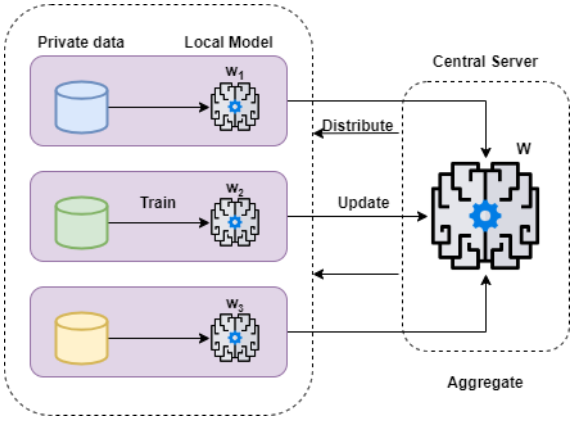


Figure 1: Visual Representation of a Typical Federated Learning Training Process

where:

$H(X)$ represents the entropy of the random variable X.

$P(x_i)$ is the probability of event x_i.

Clients with high-entropy datasets have the potential to enhance the federated learning model's performance by providing diverse and informative data.

Figure 1 depicts the standard aggregation process employed in federated learning, wherein client cooperation and server communication are established via the Federated Learning Model. The combination of data and models that make up the isolated system ensures data privacy preservation, as communication is limited to the distribution and uploading of model updates. In each isolated system, each local model is characterized by its associated weights w. When updates are received, the server uses an aggregation algorithm to obtain a global model W distributed among the clients. In this paper, "Shannon Entropy" is a criterion used for selecting clients participating in model training. The fundamental premise of our proposal lies in the synergy between entropy and the client selection process.

4 ENTROPICFL

This Section presents the system model and outlines the design of the metrics-based approach proposed in [3]. This approach forms the fundamental framework upon which our research is built. To facilitate a clear understanding of the key notations used throughout this paper, Table 1 offers a succinct summary of these symbols and terms.

4.1 Determining Update Relevance

One of the challenges in federated learning is determining the relevance and impact of each client's update on the global model. Different metrics can be used to measure the similarity or divergence between the local and global models and decide whether to send the local updates to the server. In this work, we use the following metrics:

- *Normalized Model Divergence* refers to a computational approach that quantifies the dissimilarity between local and global models by employing a particular norm. It calculates the mean value of a norm function applied to the weight disparity between client i and the global model. Specifically, when utilizing the L_1 norm, the equation is as follows:

$$dv_i = \frac{1}{|w|} \sum_{j=1}^{|w|} \left| \frac{w_{ij} - \bar{w_j}}{\bar{w_j}} \right| \tag{2}$$

  A small distance implies a close alignment between the models, while a larger distance implies a significant discrepancy.

Our approach employs the Normalized Model Divergence metric with a threshold computed based on the information clients share with the central server. This threshold is determined as a weighted average of the normalized model divergence from each client. This approach enables us to evaluate the significance of a client's update before transmitting it to the central server. By implementing this methodology, we can significantly reduce communication overhead and enhance the overall efficiency of the federated learning process.

Table 1: Notations used in this paper.

Notation	Meaning
w	Model weights
i	Index of model weights
$w_{ij}, \bar{w_j}$	$j-$th weight in client i and server, respectively
K	Server Capacity
$\mathcal{E}$	Entropy of a client
$\mathcal{A}$	Accuracy of a client
a_c	Accuracy of client c
e_c	Entropy of client c
$\|d\|$	Number of samples of client
MD	Mean of divergence

4.2 Suitability Score

Let C be a set of clients participating in a federated learning system. For each client $c \in C$, let A_c denote the accuracy, and E_c denote the entropy of client c. The suitability D_c of client c is defined as the weighted sum of its accuracy and entropy, representing its contribution to the global model, as shown in Equation 3.

$$D_c = \gamma \cdot A_c + (1 - \gamma) \cdot E_c \tag{3}$$

Where γ is a non-negative coefficient representing the relative importance of each attribute. This metric effectively assesses a client's performance by considering two crucial factors: the accuracy of its local model and entropy, indicating data diversity and richness. It is a valuable tool to evaluate individual client data quality and their contribution to the FL training process.

4.3 Details and Algorithms

EntropicFL encompasses activities that require execution on both the client and server sides. In this section, we refer to "information" on the client side as a set comprising entropy, accuracy, and Normalized Model Divergence.

Figure 2 illustrates the process of our proposed architecture. In the ①, clients who have previously received the model from the

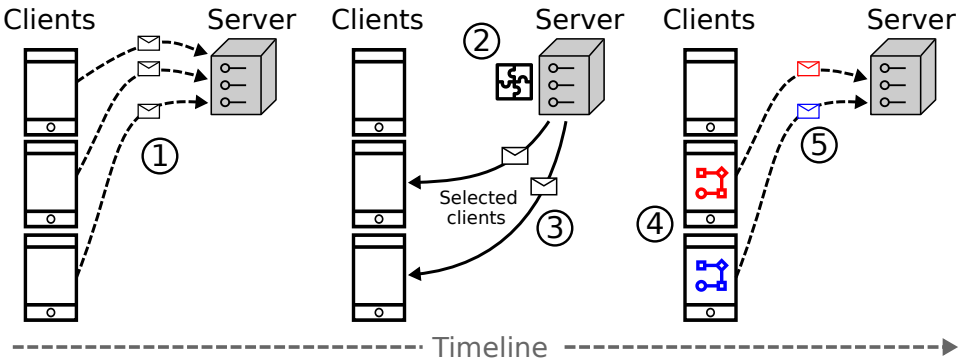


Figure 2: Overview of EntropicFL: after clients send the model in step 1, only clients selected in step 2 receive the updated model in step 3. They then proceed to train the received model in step 4. In step 5, clients train the updated model, compute the entropy of their data, evaluate the Normalized Model Divergence, and, following the criteria explained in Algorithm 2, transmit the updates.

Algorithm 1: Central Server

```
   // Initialize the model
1  foreach round t ∈ T do
2      if round == 0 then
3          w_t ← RandomInitialization( )
4          MD_t ← MeanDivergence(∞, ∞)
5          foreach client n ∈ N do
6              w^n_{t+1} ← LocalUpdate(MD_t, w_t, d_n)
7      else
8          E, A, D, |d|, w_t ← ReceiveClientInformation( )
9          MD_{t+1} ← (Σ_{i=1}^{|N|} D_i × |d_i|) / (Σ_{i=1}^{|N|} |d_i|)
           // Aggregates the gradients received
10         w_{t+1} ← Σ_{i=1}^{|N|} w^n_{t+1}
           // Client selection phase
11         ClientSelecion(E, K)
12 ClientSelection (E, K):
13     C ← NormalizeEntropy(E)
14     L ← ∅
15     for a_c, e_c ∈ C do
16         c, w ← D_c(a_c, e_c)
17         L ← L ∪ {c, w}
18     S ← ∅
19     i = 0
20     while i ≠ K do
21         c_t ← ChooseByWeights(L)
22         S ← S ∪ {c_t}
23         delete c_t from L
24         K ← K + 1
25     return S
```

Algorithm 2: Mobile Device

```
   // Receives global model w_t and mean global divergence MD_t
1  LocalUpdate (w_t, MD_t):
       // Entropy calculation using Shannon
2      E ← CalcEntropy( )
       // Compute divergence with global and client weights
3      D ← ComputeDivergence(w_t, w^c_t)
       // Accuracy
4      A ← 0
5      if client is selected then
6          w^c_t ← w_t
7          w_{t+1} ← ExecuteLocalTrain( )
8          D ← ComputeDivergence(w_t, w^c_{t+1})
9          if D ≤ MD_t OR client is priorized then
10             return E, A, D, |d|, w_t
11     return E, A, D, |d|, w_t
```

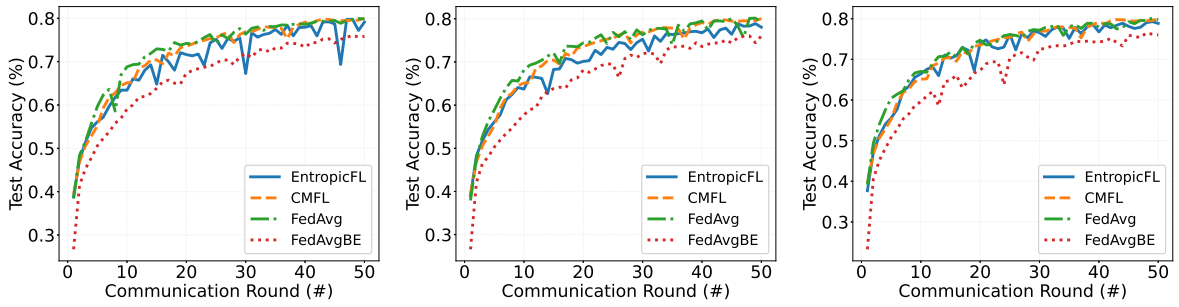
server train and communicate their updates and information to the server. This process is detailed in Algorithm 1. Then, the server receives client information accuracy, entropy, and a quantity of data in ② and calculates the Normalized Model Divergence. In ③, the server performs client selection, algorithm 1 line11, which takes the client's information and the server's capacity K as input. This algorithm normalizes entropy and uses the "ChooseByWeights" method based on Equation 3 to perform a random weighted selection, where the weight is associated with 3 for each client. Subsequently, two clients from the selected group are prioritized randomly. Finally, in ④, clients execute Algorithm 2, and in step ⑤, the clients calculate the entropy of its data and assess the Normalized Model Divergence between the weights communicated by the server and the local weights obtained after training using Equation 2. Subsequently, they transmit their updates (line 9).

5 EXPERIMENTAL RESULTS

In this section, we present empirical evaluations of EntropicFL, along with comparative assessments against other federated learning methods.

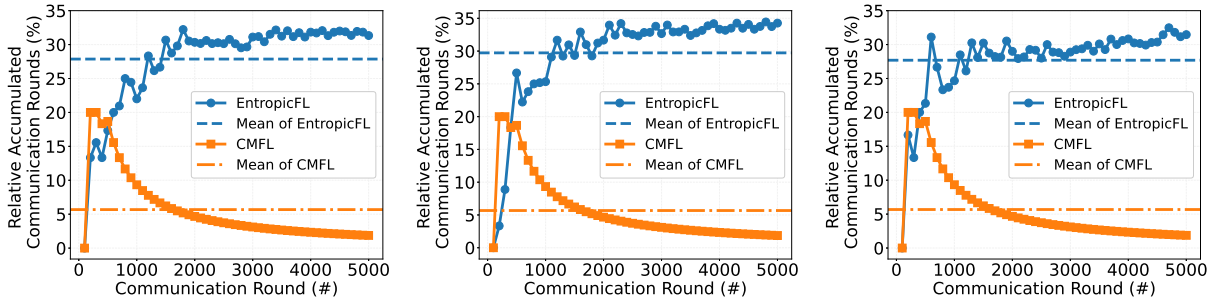
5.1 Performance Metrics

To facilitate the comparison between the algorithms, the following metrics were employed. Within the f^{th} iteration, we utilize the



(a) Accuracy for $\gamma = 0$ (b) Accuracy for $\gamma = 0.5$ (c) Accuracy for $\gamma = 1$

Figure 3: Accuracy results with different γ values in Equation 3.



(a) Communication overhead reduction for $\gamma = 0$ (b) Communication overhead reduction for $\gamma = 0.5$ (c) Communication overhead reduction for $\gamma = 1$

Figure 4: Communication overhead reduction results with different γ values compared to the FedAvg baseline.

notation S_t to represent the ensemble of clients that transmit their local updates to the central server. The nomenclature *communication round* in the t^{th} iteration is formally defined as $r_t = |S_t|$. The notion of *accumulated communication rounds* is established as the cumulative tally of local updates performed by clients over a period spanning T iterations using an algorithm A:

$$\Phi_A^a = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} |S_t| \tag{4}$$

As demonstrated in previous studies, as exemplified by [17], the primary aim is to devise a communication-efficient algorithm denoted as A with the goal of minimizing the accumulated communication rounds, denoted as Φ_A^a, while ensuring a specific level of learning accuracy denoted as a. We use the same notation and objective.

5.2 Experiments Setup

EntropicFL utilized Python version 3.10 and the Flower framework [2]. The architecture employed is a Convolutional Neural Network (CNN), specifically, the AlexNet architecture as described in [7].

We generated non-uniform data distribution to FL clients using the CIFAR-10 dataset and compared EntropicFL with literature approaches namely CMFL [17], FedAvg [13], and FedAvgBE [15]

This was accomplished by implementing a data distribution model founded on the Dirichlet distribution with a specified parameter value of $\alpha = 0.1$, resulting in the creation of a non-IID data distribution. Within the simulation environment, we simulated the 30 clients while the server's capacity was restricted to 15 to emulate the scenario with constrained capacity edge nodes supporting the FL training/aggregation process.

The use of the default threshold determined by the CMFL model is not feasible due to the observed variability in metric values stemming from differences in dataset configurations among various clients. To address this issue, we have chosen to employ CMFL with a manually selected threshold value of 0.9991. Additionally, we have configured FedAvgBE to use a fixed local batch size of 2 and set the number of training local epochs to a consistent value of two for all clients.

5.3 Results

Along with the execution of the experiments, the accuracy of EntropicFL closely rivals that of the FedAvg method throughout the Federated Learning process. Nevertheless, a notable distinction arises from the observed performance decline in specific rounds. This decline is attributed to the client selection process and the model divergence assessment, which guides each client's decision to share their model weights with the server. In our proposed method, EntropicFL, configured as described, we effectively mitigate the observed 'communication overhead' as depicted in Figure 3(a), 3(b), and 3(c), with configurations for γ set to 0, 0.5, and 1. When the

parameter γ is set to 0, the metric outlined in equation 3 takes exclusive precedence within the selection criteria, with entropy emerging as the dominant factor. The accuracy performance resembles that achieved when γ remains fixed at 1.

We consistently observed that adjusting the γ parameter does not substantially impact accuracy. Surprisingly, it yields accuracy levels closely mirroring those of the traditional FedAvg method. However, the remarkable aspect of this consistent accuracy is the significant reduction in communication overhead costs. These findings underscore the potential value of the γ parameter, especially in scenarios where data exhibits temporal variations and lacks consistency across training rounds.

This paper has introduced the use of entropy as a client selection criterium in federated learning and combined with a mechanism to select if the client sends or not send updates to the server. The preliminary results show that improvements can be achieved in terms of communication, potentially reducing bottlenecks and also potentially reducing energy consumption in the edge devices.

Further investigation is needed in order to have more generalized conclusions in a variety of scenarios, as well as to investigate how to combine entropy and other additional criteria, besides model divergence, in the client selection process.

6 CONCLUSION AND FUTURE WORK

In this paper, we have developed and evaluated a federated learning framework that has proven to be effective in reducing communication overhead. Our solution has significantly decreased this metric without compromising the model's performance to a great extent, maintaining a very similar accuracy to the FedAvg approach.

We have introduced a metric that combines entropy and model accuracy, enriching our evaluation and enabling more informed decision-making. Throughout our experiments, we observed accumulated communication rounds values of approximately 27.7%, 27.9%, and 29.7% for the configurations of $\gamma = 1$, $\gamma = 0$, and $\gamma = 0.5$, respectively. These results highlight that the $\gamma = 0.5$ configuration achieved the greatest reduction in the number of accumulated communication rounds, suggesting its effectiveness in our specific context, influenced by the characteristics of our data.

For future work, we plan to delve deeper into the relationship between accuracy and entropy, conduct validations with more complex datasets and a larger number of clients, and explore techniques such as quantization and model pruning to further reduce communication overhead, both on the client and server sides. Additionally, we will consider incorporating more advanced client selection strategies and adapting our framework to more challenging federated learning environments.

ACKNOWLEDGEMENTS

The authors would like to thank to PPI-Softex with support from the MCTI [01245.013778/2020-21]. This work has been partially funded by CNPq.

REFERENCES

[1] Sawsan Abdulrahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi. 2021. FedMCCS: Multicriteria Client Selection Model for Optimal IoT Federated Learning. *IEEE Internet of Things Journal* 8, 6 (2021), 4723–4735. https://doi.org/10.1109/JIOT.2020.3028742

[2] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. 2020. Flower: A Friendly Federated Learning Research Framework. *CoRR* abs/2007.14390 (2020). arXiv:2007.14390 https://arxiv.org/abs/2007.14390

[3] Lei Fu, Huanle Zhang, Ge Gao, Mi Zhang, and Xin Liu. 2023. Client Selection in Federated Learning: Principles, Challenges, and Opportunities. *IEEE Internet of Things Journal* (2023), 1–1. https://doi.org/10.1109/JIOT.2023.3299573

[4] Badra Souhila Guendouzi, Samir Ouchani, Hiba EL Assaad, and Madeleine EL Zaher. 2023. A systematic review of federated learning: Challenges, aggregation methods, and development tools. *Journal of Network and Computer Applications* (2023), 103714. https://doi.org/10.1016/j.jnca.2023.103714

[5] Huawei Huang, Ruixin Li, Jialiang Liu, Sicong Zhou, Kangying Lin, and Zibin Zheng. 2022. ContextFL: Context-aware Federated Learning by Estimating the Training and Reporting Phases of Mobile Clients. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. 570–580. https://doi.org/10.1109/ICDCS54860.2022.00061

[6] Peter Kairouz and H. Brendan et al. McMahan. 2021. *Advances and Open Problems in Federated Learning*. arXiv. https://arxiv.org/abs/1912.04977

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[8] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient Federated Learning via Guided Participant Selection. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*. USENIX Association, 19–35. https://www.usenix.org/conference/osdi21/presentation/lai

[9] Na Lin, Yamei Wang, Enchao Zhang, Keping Yu, Liang Zhao, and Mohsen Guizani. 2023. Feedback delay-tolerant Proactive Caching Scheme Based on Federated Learning at the Wireless Edge. *IEEE Networking Letters* 5, 1 (2023), 26–30.

[10] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. 2022. A state-of-the-art survey on solving non-IID data in Federated Learning. *Future Generation Computer Systems* 135 (2022), 244–258. https://doi.org/10.1016/j.future.2022.05.003

[11] Filipe Maciel, Allan M De Souza, Luiz F Bittencourt, and Leandro A Villas. 2023. Resource Aware Client Selection for Federated Learning in IoT Scenarios. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 1–8.

[12] Priyanka Mary Mammen. 2021. Federated Learning: Opportunities and Challenges. arXiv:2101.05428 [cs.LG]

[13] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs.LG]

[14] Aissa Hadj Mohamed, Nícolas RG Assumpção, Carlos A Astudillo, Allan M de Souza, Luiz F Bittencourt, and Leandro A Villas. 2023. Compressed Client Selection for Efficient Communication in Federated Learning. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*. IEEE, 508–516.

[15] Fernanda C. Orlandi, Julio C. S. Dos Anjos, Valderi R. Q. Leithardt, Juan Francisco De Paz Santana, and Claudio F. R. Geyer. 2023. Entropy to Mitigate Non-IID Data Problem on Federated Learning for the Edge Intelligence Environment. *IEEE Access* 11 (2023), 78845–78857. https://doi.org/10.1109/ACCESS.2023.3298704

[16] Osama Shahid, Seyedamin Pouriyeh, Reza M. Parizi, Quan Z. Sheng, Gautam Srivastava, and Liang Zhao. 2021. Communication Efficiency in Federated Learning: Achievements and Challenges. arXiv:2107.10996 [cs.LG]

[17] Luping WANG, Wei WANG, and Bo LI. 2019. CMFL: Mitigating Communication Overhead for Federated Learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 954–964. https://doi.org/10.1109/ICDCS.2019.00099

[18] Chengxu Yang, Mengwei Xu, Qipeng Wang, Zhenpeng Chen, Kang Huang, Yun Ma, Kaigui Bian, Gang Huang, Yunxin Liu, Xin Jin, and Xuanzhe Liu. 2022. FLASH: Heterogeneity-Aware Federated Learning at Scale. *IEEE Transactions on Mobile Computing* (2022), 1–18. https://doi.org/10.1109/TMC.2022.3214234

[19] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775. https://doi.org/10.1016/j.knosys.2021.106775

[20] Wenyu Zhang, Xiumin Wang, Pan Zhou, Weiwei Wu, and Xinglin Zhang. 2021. Client Selection for Federated Learning With Non-IID Data in Mobile Edge Computing. *IEEE Access* 9 (2021), 24462–24474. https://doi.org/10.1109/ACCESS.2021.3056919

[21] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. *arXiv* (2018). https://doi.org/10.48550/ARXIV.1806.00582