



Eco-FL: Enhancing Federated Learning sustainability in edge computing through energy-efficient client selection

Martina Savoia, Edoardo Prezioso, Valeria Mele, Francesco Piccialli *

University of Naples Federico II, Department of Mathematics and Applications “R. Caccioppoli”, Italy

ARTICLE INFO

Keywords:

Federated learning
Edge cloud-computing
Sustainability
Energy consumption
Client selection

ABSTRACT

In the realm of edge cloud computing (ECC), Federated Learning (FL) revolutionizes the decentralization of machine learning (ML) models by enabling their training across multiple devices. In this way, FL preserves privacy and minimizes the need for centralized data by processing data near the source. From a communication standpoint, only the model weights are exchanged between devices. By avoiding the need to send data to a centralized location for processing, FL reduces the energy required for data transfer and supports more efficient use of computing resources at the edge. FL is particularly advantageous for resource-constrained devices, such as smartphones and IoT devices. However, this limited computational power and battery capacity and the challenge of energy consumption are critical aspects of FL systems. This paper introduces Eco-FL, an innovative methodology designed to optimize energy consumption in FL systems, in the field of Green Edge Cloud Computing (GECC). Our approach employs a device selection process that considers the entropy of the data held by the devices and their available energy reserves. This ensures that devices with lower energy availability are less likely to participate in the training rounds, prioritizing those with higher energy capacities. To evaluate the efficacy of our methodology, we utilize FedEntropy, an entropy-based aggregation method, alongside established aggregation methods such as FedAvg and FedProx for performance comparison. The effectiveness of Eco-FL in reducing energy consumption without compromising the accuracy of the FL process is demonstrated through analyses conducted on three distinct datasets. These analyses vary the β parameter of the Dirichlet distribution and account for scenarios with both homogeneous and heterogeneous initial device charges. Our findings validate Eco-FL’s potential to enhance the sustainability of FL systems by judiciously managing client participation based on energy criteria, presenting a significant step forward in the development of energy-efficient FL.

1. Introduction

Edge Cloud-Computing (ECC) has emerged as a transformative paradigm in the realm of information technology, bringing computational capabilities closer to the data source [1]. Unlike traditional cloud computing, which centralizes data processing in distant data centers, ECC distributes computing resources to the “edge” of the network. This proximity to data sources reduces latency, enhances real-time processing, and accommodates the requirements of applications that demand low-latency responses [2]. Federated Learning (FL) was first introduced in [3] and is characterized by its distributed model training on local devices, representing a paradigm shift in machine learning (ML) and aligning seamlessly with the principles of ECC. FL begins with the creation of a global model, which is then shared across all participating devices and serves as the starting point for training. Each device refines the global model using its locally stored data. This

ensures that learning occurs without the need to share raw data, a critical aspect in privacy-conscious applications [4]. Following local training, devices communicate only the model updates, typically in the form of weight adjustments, to a central server. This process is iterative, with constant refinement based on contributions from diverse data sources. Each iteration of the training process is commonly referred to as a “round”. This cyclic process continues until the model achieves the desired performance or the number of allowed rounds ends.

The main challenge of FL is in managing data privacy and security. As data remains locally on client devices and the model is updated via wireless communications, it is essential to ensure that sensitive data is not exposed to risks during the learning process. Hence, FL must address device and network heterogeneity, ensuring that the aggregated model is representative and generalizable across a wide range of devices. Another challenge is efficiently synchronizing distributed

* Corresponding author.

E-mail address: francesco.piccialli@unina.it (F. Piccialli).

models. Nevertheless, one of the challenges posed by FL's decentralized approach is the issue of energy consumption, especially in situations where devices are heterogeneous and have limited resources, such as smartphones. Moreover, in FL, data is typically collected from a diverse set of clients, each with its own unique data distribution, leading to a non-identically distributed (non-IID) nature of data across devices that can impact the FL process [5]. To enhance sustainability, this study aims to:

- propose a new client selection method based on data entropy and residual energy available to each client. In this way, it is possible to search for a trade-off between data entropy, crucial for achieving good performance, and energy consumption.
- introduce a new aggregation method, using only weights from selected clients, after training.
- evaluate our Eco-FL framework on three datasets and compare it with other FL methods, in terms of accuracy and consumed energy.

The work is organized as follows: in Section 2 we present related works, in Section 3 we present Eco-FL, providing details on the chosen optimization method to select clients. In Section 4, we introduce the hardware infrastructure utilized, the datasets employed (CIFAR-10, CIFAR-100, and CINIC-10), the model design, and all aspects of the experimentation. In Section 5 we showcase the results obtained on three datasets and perform a comparison with FedEntropy and common aggregation methods such as FedAvg [6], useful for its adaptability to unbalanced and non-IID data distributions and FedProx [7], that can be seen as an extension and reconfiguration of FedAvg. Additionally, this section presents results related to various distributions of available energy, their respective metrics, and tests that simulate device malfunctions, thus approaching a more realistic approach in the context of FL. Lastly, in Section 6 conclusions are provided.

2. Related works

Federated Learning (FL) and High-Performance Computing (HPC) are closely related in the field of Edge Cloud-Computing (ECC). FL, which leverages a distributed approach for training models on local data from edge devices, can be effectively facilitated by using an HPC infrastructure to handle vast amounts of data and computationally intensive tasks. HPC can leverage multicore architectures to tackle complex problems efficiently [8,9]. Problems associated with HPC include resource contention, high energy consumption, complex programming paradigms and fault tolerance [10]. FL finds application in various domains, including healthcare [11,12], finance [13], smartphones [14], and Internet of Things (IoT) [15], facilitating collaborative model training without compromising the integrity of sensitive information. Especially in the so-called Internet of Medical Things (IoMT), ensuring privacy is fundamental. Zhou et al. [16] introduces the Federated Distillation and Blockchain empowered Secure Knowledge Sharing (FDBC-SKS) model, which addresses data sharing in IoMT environments by transforming it into a collaborative model knowledge sharing problem. This lightweight distributed deep learning framework utilizes a peer-to-peer federated distillation mechanism for decentralized FL, enhancing model flexibility and reducing communication consumption. Decentralized FL, enables direct communication between clients and avoid the need of transfer parameters to a central server, which may be vulnerable to malicious network attacks [17]. Zhou et al. [18] presents a Privacy-Perceiving Asynchronous Federated Learning (PPAFL) framework that is built on peer-to-peer (P2P) and aims to accomplish decentralized model training for secure and robust modern mobile robotic systems in networks that go beyond 5G. Using the large-scale, heterogeneous, and multi-modal Metaverse data, Zhou et al. [19] present a Personalized Federated Learning with Model-Contrastive Learning (PFL-MCL) architecture that may effectively improve the communication and interaction in human-centric Metaverse environments. Zhou

et al. [20] suggests using an edge-cloud Digital Twin (DT) system in conjunction with a three-layer Federated Reinforcement Learning (FRL) framework to achieve lightweight and communication-efficient model training and improve real-time processing for the deployment of intelligent applications, particularly in high-speed mobile networks. Moreover, FL can be used for implementation in real-time and critical life situations, as well as for simplifying management in smart city applications [21]. In this context, Intelligent Transportation System (ITS) needs real-time, accurate, and private location data. In [22], a Spatial-Temporal Federated Transfer Learning (ST-FTL) framework is introduced to improve cooperative GPS positioning accuracy in urban ITS while ensuring data privacy. Transfer Learning (TL) is employed to optimize the global model's initialization, resulting in faster convergence and reduced communication costs in FL. Global model synchronization is a challenge in the FL domain, addressed by Hierarchical Federated Learning (HFL), which aggregates models of nearby devices at edge servers and periodically synchronizes them via the cloud. However, devising an efficient synchronization strategy for HFL remains a significant challenge, particularly due to factors like device diversity, non-IID data, and device mobility [23]. An Adaptive Segmentation enhanced Asynchronous Federated Learning (AS-AFL) model is presented in [24] to address issue related to synchronous FL. This framework uses a decentralized implementation of FL to attain edge intelligence in sustainable computing systems.

FL can be widely utilized on battery-powered devices, such as smartphones, as they generate data that requires privacy preservation [25,26]. Furthermore, it is important to consider that in reality, devices may have intermittent energy during training [27] and limited resources.

When numerous clients are available, selecting clients is crucial to avoid excessively high latency or to reduce energy consumption. Moreover, FL is susceptible to malicious that share poor-quality data, lowering the overall quality of the model. Hence, selecting clients is crucial to ensure the accuracy and efficiency too. In the context of IoMT, Zhou et al. [28] proposes a clustering-based method to choose participants using social context information. Group-specific FL is carried out after various edge participant groups have been formed. The global model's robustness is enhanced by further aggregating the models of the different edge groups. Generally, assuming all clients have good quality data, the higher the number of clients, the better the performance of the global model. Therefore, often there is an attempt to find a trade-off between performance and the number of clients chosen, to reduce energy consumption. Albelaihi et al. [29] aims to optimize this trade-off by maximizing the number of selected clients and minimizing battery energy consumption, ensuring that all selected clients have sufficient energy to load their local models before the deadline. Client selection usually involves solving an optimization problem. Xu and Wang [30] explores the impact of different client selection strategies (ascending, descending, uniform) on performance, addressing an optimization problem for both client selection and bandwidth management under energy constraints. They highlight the significant influence of the timing of client participation on the overall performance of the FL process. Albelaihi et al. [31] extends beyond just computational and uploading latency considerations to incorporate waiting time for participant selection. The aim is to optimize the selection process to maximize the number of participants able to upload their local models before the deadline within a global iteration. The paper introduces the Latency aware Participant selection algorithm (LEARN) to address this optimization problem. Moreover, the training process can become inefficient when clients have limited computational resources (i.e., requiring longer update time) or operate under poor wireless channel conditions, resulting in longer upload time. The FedCS protocol [32], mitigates this issue and efficiently performs FL by actively managing clients based on their resource conditions. Ling et al. [33] develops FedEntropy, which computes the entropy of soft labels sent to the server after the initial training, categorizing clients into "positive" and

“negative”. Positive clients increase entropy and are selected to send their local models for aggregation. Ribeiro and Vikalo [34] propose a framework for updating the global model in systems constrained by communication, by selectively requesting input only from clients providing informative updates, and estimating the local updates that are not communicated.

Various other phases of the FL process can be analyzed to reduce energy consumption. Wang et al. [35] utilizes data as a flexible parameter to adjust training schedules, aiming to achieve nearly optimal solutions for computation time and accuracy loss. Building upon offline profiling, they define optimization problems and introduce polynomial-time algorithms applicable to both class-balanced and unbalanced data scenarios. In [36] FL parameters are evaluated to minimize carbon emissions while maintaining good performance and training time, in both synchronous and asynchronous FL scenarios. Kim et al. [37] solves a multi-objective optimization problem to reduce the number of communication rounds and energy consumption while achieving a target accuracy by developing a Quantum Neural Network (QNN) for each device. Salh et al. [38] explores ways to reduce latency and energy consumption for computations and communication by resources without compromising accuracy. Salh et al. [39] resolves the bandwidth optimization problem to reduce energy consumption on IoT devices. Li et al. [40] develops energy consumption control using gradient reduction by devices through sparsification, quantization, encoding and element-wise aggregation by the server, named FedGreen. Additionally, an exploration of different compression ratios and trade-offs between accuracy and energy efficiency is proposed. FedGreen allows various devices to compress local gradients upon request, based on their energy states. In [41], the interaction between edge servers and cloud servers is explored, to optimize system energy consumption while adhering to service latency constraints. Zheng et al. [42] utilizes entropy theory to measure the difference between different local model parameters and proposes an adaptive learning rate for each client via a mean-field approach, which effectively estimates the terms related to other clients' model parameters over time and avoids frequent communication. Sun et al. [43] propose an online scheduling policy that is energy-aware and dynamic, aiming to maximize the average number of workers scheduled for gradient updates per iteration while adhering to a long-term energy constraint.

Our work proposes a novel approach that considers energy optimization within the context of dynamic worker scheduling, enabling more efficient resource management. We introduced an innovative criterion for client selection that balances data entropy and available residual energy, for each device. Our goal, from a sustainability perspective, is to ensure that the achieved accuracy is preserved while consuming less energy.

3. Methodology

Client selection can influence the FL performance, in terms both of accuracy and energy consumption. Typically, to reduce energy consumption, there is a need to find a trade-off between accuracy and energy consumption. Hence, the core of the proposed methodology focuses on solving an optimization problem to select clients efficiently, based on the following two key aspects:

- **Residual energy:** clients are evaluated in terms of their remaining energy reserves. The optimization problem aims to select devices with the highest residual charge. Devices with excessively low residual energy are excluded from participation to prevent potential disruptions due to energy depletion during the FL process. This safeguards against the selection of clients that may not sustain the computational requirements of model training.
- **Data entropy:** in information theory, entropy is a measure of uncertainty or randomness associated with a random variable. It quantifies the amount of information contained in a message, event or random variable.

Each client may carry different types and amounts of information, based on how datasets are partitioned among them. Clients are assessed based on the entropy of their local datasets. High data entropy signifies diverse and information-rich datasets, making the client a valuable contributor to the FL model. This criterion ensures that clients with meaningful and diverse data are given priority in the selection process. This is crucial to ensure that the global model adapts as closely as possible to the diverse data.

Clients with devices having higher available energy and entropy are prioritized to maximize the utilization of the available resources. Additionally, in the initial training rounds, only a few clients are selected, as demonstrated in [30], as few clients at the beginning of FL processes suffice to achieve good model weights, which are then updated in subsequent rounds.

For the i th client, $E_i = -\sum_{j=1}^m p_{i,j} \log(p_{i,j})$ represents the data entropy, m is the number of classes present in the client data, x_i is a binary variable indicating if the client is selected and w_i denotes the residual energy. If n represents the total number of clients, R_{\max} denotes the total number of rounds and r is the current round, the developed optimization problem is reported in Eq. (1).

Optimization Problem

$$\begin{aligned}
 \text{Maximize: } & \sum_{i=1}^n x_i [\alpha w_i + (1 - \alpha) E_i] \\
 \text{Subject to: } & \sum_{i \in C} x_i \leq k + r & (1a) \\
 & \sum_{i \in C} x_i \leq n \cdot 0.1 & (1b) \\
 & \sum_{i \in C} x_i \geq 2 & (1c) \\
 & 1 \leq r \leq R_{\max} & (1d) \\
 & w_i > 0 & (1e) \\
 & x_i \in [0, 1] & (1f) \\
 & \alpha \in 0, 1 & (1g)
 \end{aligned} \tag{1}$$

Constraint (1a) limits the number of selected devices to the minimum number plus the current round number. This approach allows the upper limit of selectable devices to increase as the process progresses. Constraint (1b) limits the maximum number of selectable devices to 10% of the total. Constraint (1c) enforces a lower limit on the number of devices to be selected, set as 2. α represents the weight to be assigned to energy and entropy within the objective function. Considering $\alpha = 1.0$ corresponds to the particular case where only energy is used as a criterion in client selection. Due to the difference in scale between energy consumption, typically expressed in Wh (Watt-hour), and entropies, initially the optimization problem gives less importance to data and more to consumption. As the available energy decreases, the two quantities become comparable. The Eco-FL architecture is visually depicted in the schematic representation reported in Fig. 1. In this architecture, a central server maintains communication with individual clients, each characterized by unique levels of data entropy and available energy resources, which are relayed to the server. The server orchestrates the selection of clients designated for training tasks, using the optimization problem (1). Upon completion of each round, participating clients transmit their respective local parameters back to the server. Subsequently, the server aggregates these parameters and proceeds to evaluate the model across all clients, followed by updating the overall energy consumption status. To quantify the energy consumption associated with training on GPUs, we sample the energy consumption during the training by querying the NVIDIA System Management Interface repeatedly. By multiplying the energy consumption obtained at each instant by the duration of each respective operation

and summing these values we calculate the training consumption for each client (see Eq. (2))

$$C_i = \sum_{j=1}^m w_j \cdot t_j \quad (2)$$

where C_i is the client i 's energy consumption, w_j is the j th consume for the j th operation and t_j is the relative time to do the j operation. Hence, the total energy consumption for the training FL process is

$$C = \sum_{i=1}^n C_i \quad (3)$$

where n is the total number of clients. In this phase, the energy consumption of clients that are not selected to train but remain idle and only conduct global model testing is also taken into account.

Algorithm 1 Implementation Eco-FL

Input: n = # of clients, R_{\max} = # of training rounds, D_i dataset i -th client, S_d set of selected devices, AE Available Energy (Wh), CE Consumed Energy (Wh).

```

1:  $dict \leftarrow \{ \}$ 
2: for  $i = 1, \dots, n$  do
3:    $dict[i] \leftarrow GetProperties(D_k, AE)$ 
4:    $entropy = list(dict[i]['entropy'])$  for  $i = 1, \dots, n$ 
5: for  $r = 1, \dots, R_{\max}$  do
6:    $AE = list(dict[i]['energy'])$  for  $i = 1, \dots, n$ 
7:    $S_d \leftarrow \text{Optimization Problem (1)}$ 
8:   for  $s \in S_d$  do
9:     Train and evaluate model, compute  $CE$ 
10:     $AE \leftarrow AE - CE$  for  $i = 1, \dots, n$ .
11:   for  $s \notin S_d$  do
12:     Evaluate model, compute  $CE$ 
13:      $AE \leftarrow AE - CE$  for  $i = 1, \dots, n$ .
14: Accuracy  $\leftarrow$  Test model on common test set
Output: Accuracy

```

4. Experimental setup

4.1. Flower framework

To manage the FL process and establish a collaborative environment between a server and multiple clients, the Flower framework was utilized [44]. Flower is known for its ability to extend FL implementations to mobile and wireless clients, accommodating a range of computational, memory, and network resources and its adaptability in incorporating emerging algorithms, training strategies, and communication protocols. Eco-FL and the benchmark methods were implemented in the Flower framework.

4.2. Eco-FL algorithm description

Algorithm 1 presents a schematization of the Eco-FL procedure. The algorithm begins by initializing the necessary input parameters. These include the number of clients n , the maximum number of training rounds R_{\max} , the dataset D_i of the i th client, the set of selected devices S_d , the available energy AE in watthours (Wh), and the consumed energy CE in watthours (Wh). An empty dictionary $dict$ is initialized to store the properties of each client device (line 1). The algorithm then loops through each client to collect their properties (lines 2–5). For each client, the $GetProperties(D_k, AE)$ function is called to retrieve and store properties such as available energy (AE) and dataset entropy ($entropy$) in the $dict$ (line 3). A list of entropies for all clients is created using the collected properties. This is done using a list comprehension that iterates over the dictionary to extract the entropy values (line 4). The algorithm then enters a loop for each training round (lines 5–13).

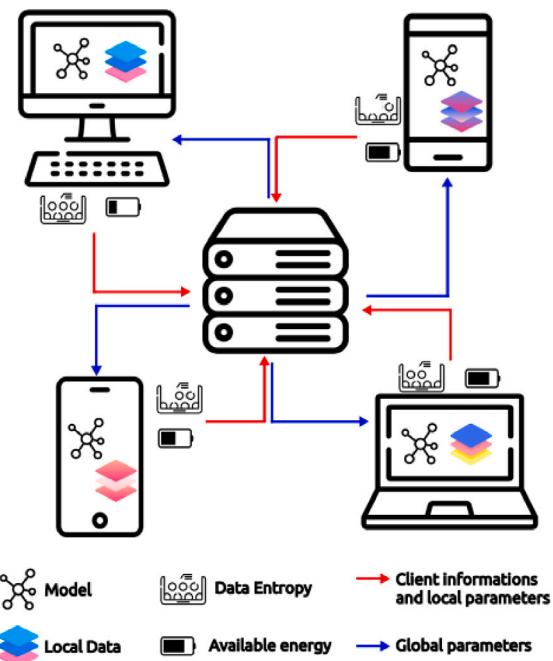


Fig. 1. Illustration of the Eco-FL architecture, depicting a central server interacting with clients, each characterized by distinct levels of data entropy and available energy. Clients relay their attributes to the server, which selects participants for training rounds. Upon completion of a round, clients transmit their local parameters back to the server, which aggregates them and evaluates the model across all clients. Finally, the available energy for each client is updated based on the consumption of the round.

At the beginning of each round, the list of available energy for all clients is updated using the values stored in $dict$. This is done using a list comprehension that iterates over the dictionary to extract the energy values (line 6). The set of devices S_d that will participate in training is determined by solving an optimization problem that takes into account the energy constraints and the entropy of the datasets (line 7). For each selected device in S_d , the algorithm trains and evaluates the local model, computing the consumed energy CE (line 9). The available energy AE is then updated by subtracting the consumed energy CE for each client (line 10). The algorithm also evaluates the local models of the non-selected devices, computing the consumed energy CE without active training (line 12). The available energy AE is updated accordingly (line 13). After completing the training and evaluation for all devices, the global model is tested on a common test set to determine its accuracy (line 14). The final accuracy of the global model is then outputted.

4.3. Hardware infrastructure

The experiments were conducted on a multi-node server setup to accurately measure the electrical consumption of each GPU across different clients, ensuring that the measurements were not influenced by concurrent operations on the same GPU. For this purpose, we utilized the Infrastructure for Big Data and Scientific Computing (I.B.I.S.CO) provided by the S.Co.P.E. Data Center at the University of Naples Federico II. This infrastructure offers a heterogeneous environment that aligns well with our requirements [45]. The system comprises 36 nodes, with 32 of them dedicated to computing and the remaining four to storage. Each computing node is equipped with a DELL C4140 server, which houses 4 NVIDIA Tesla V100 GPUs, summing up to a total of 128 GPUs across all computing nodes. We tailored the maximum capacity based on the maximum consumption of each GPU, which is 250 W.

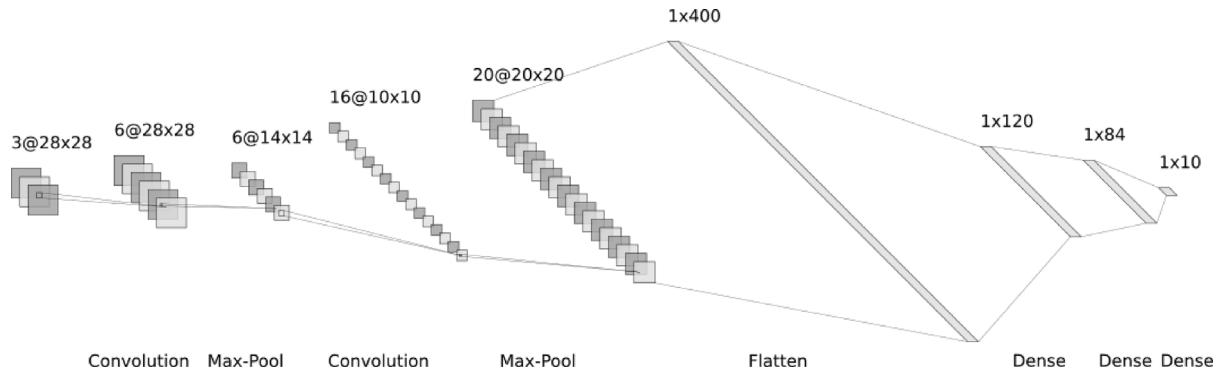


Fig. 2. The architecture of the LeNet model for image recognition. The model consists of two convolutional layers followed by pooling layers and three Dense layers. The first convolutional layer has 6 filters of size 5×5 , while the second has 16 filters of size 5×5 . Both pooling layers use a pool size of 2×2 . The Dense layers consist of two layers with 120 and 84 neurons respectively, followed by an output Dense layer with the number of classes in the dataset. Note that the last layer has dimensions 1×20 for CIFAR-100 and 1×10 for CIFAR-10 and CINIC-10. The model size is 0.24 MB.

4.4. Dataset

For the experimentation, CIFAR-10, CIFAR-100 [46], and CINIC-10 [47] datasets were utilized for image classification. Note that for CIFAR-100, we use 20 superclasses of labels. For experiments in the non-IID setting, we opted to utilize various dataset distributions. In particular, we selected the Dirichlet distribution, with β equal to 0.1, 0.3, 0.5 and 1.0. Note that when β is small, the distribution is more spread out, resulting in a diverse dataset distribution across clients. As β increases, the distribution becomes more concentrated around specific outcomes, leading to a more homogeneous dataset distribution.

4.5. Homogeneous and heterogeneous energies

Homogeneous devices exhibit uniform characteristics across the FL network, typically in terms of computational power, memory capacity, and energy resources, while heterogeneous devices vary significantly in their computational capabilities and energy availability. Generally, the diversity of heterogeneous devices offers potential benefits such as enhanced model robustness but also introduces challenges related to resource management, communication overhead, and model convergence. In our experimental setup, we explore both scenarios regarding starting energies: one where the maximum energy consumption is uniform across all clients, and another where it is heterogeneous. In the heterogeneous scenario, each client is assigned a value derived from a Gaussian probability distribution, which remains fixed for all experimental cases.

4.6. Model and hyperparameters

We employed the LeNet model, which consists of two convolutional layers followed by max-pooling layers and three Dense layers. The first convolutional layer takes an input of size $3 \times 32 \times 32$ (RGB images) and applies 6 filters of size 5×5 , resulting in feature maps of size $6 \times 28 \times 28$. The second convolutional layer takes the feature maps from the previous layer and applies 16 filters of size 5×5 , producing feature maps of size $16 \times 10 \times 10$. Max-pooling layers follow each convolutional layer, reducing the spatial dimensions of the feature maps by half while retaining the most important information. After the convolutional layers, the feature maps are flattened and passed through two Dense layers with 120 and 84 neurons, respectively. Finally, the output layer consists of a Dense layer with the number of neurons equal to the number of classes in the dataset (see Fig. 2). The *ReLU* activation function is used after each convolutional and Dense layer, except for the output layer. The model size is 0.24 MB and the chosen optimizer is *SGD*, with a learning rate of 0.01 and a momentum of 0.5.

5. Results

Experiments were conducted on three datasets (CIFAR-10, CIFAR-100 and CINIC-10) in non-IID scenarios using various data distributions. We choose 100 clients and 1500 rounds with 5 local epochs for each client to run FL. The fixed maximum energy for all clients is set to 15 Wh for the homogeneous case, while for the heterogeneous case, each client is assigned a value obtained through a Gaussian probability distribution centered in 15 Wh with a standard deviation of 2 Wh. Additionally, we set the target accuracy threshold at 40%. If the target accuracy is reached before reaching the maximum number of rounds, an additional 20 communication rounds are performed before the process is interrupted. This allows us to compare the energy consumption required to achieve the target accuracy, ensuring a fair comparison of performance.

Our methodology was compared with FedEntropy, which utilizes client entropy for selection, as well as with FedAvg and FedProx, where client selection occurs randomly. The number of selected clients is always 10% of the available clients for FedAvg, FedProx, and FedEntropy, while for Eco-FL, it is lower or equal to 10% of the clients. This is because a gradually increasing upper limit from 2% to 10% was imposed on the number of selectable clients, which increases as the number of rounds progresses. In this way, our methodology selects as few clients as possible, reducing energy consumption, especially in the early training rounds. Indeed as seen in [30], the timing of client participation significantly influences the ultimate performance of the FL process.

5.1. Ablation study

In this section, we present the analysis aimed at evaluating the performance of our proposed methodology under various experimental conditions. Our analysis focuses on examining how key parameters, such as the β parameter in the Dirichlet distribution and the heterogeneity of energy availability among devices, impact the effectiveness of FL processes. By varying these parameters across multiple datasets (CIFAR-10, CIFAR-100, CINIC-10) and FL strategies, we highlight the robustness of our approach. Furthermore, we experimented to assess how the performance varies with the α parameter in the objective function of the optimization problem, which weights data entropy and energy.

5.1.1. Devices with homogeneous initial energy availability

Firstly we show our experimental results using fixed starting available energies for all clients and the parameter α set to 0.5 in the optimization problem. Results for each of the three datasets, varying β and the client selection method are reported in Table 1.

Table 1

Total Wh consumed with homogeneous initial energy availability. The method with the lowest energy consumption, given the same β and dataset, is highlighted in bold.

Dataset	β	Eco-FL	FedEntropy	FedAvg	FedProx
CIFAR-10	0.1	227	1143	6923	13 433
	0.3	200	180	3763	1215
	0.5	194	95	2541	5135
	1.0	91	295	1443	4571
CIFAR-100	0.1	1583	7285	29714	10 634
	0.3	1546	2131	64 407	7966
	0.5	1536	1923	19 661	7237
	1.0	1524	8286	68 291	3644
CINIC-10	0.1	1471	1223	63 519	6995
	0.3	637	2371	26 675	19 751
	0.5	647	660	15 836	4944
	1.0	550	602	2673	10 182

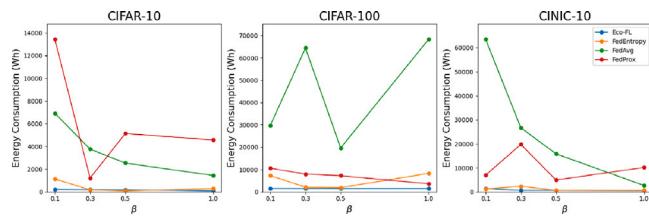


Fig. 3. Energy consumption for various methods and dataset, varying β .

Comparing the energy consumption reported in the table, our strategy proves to be better in terms of energy consumption compared to the other 3 methodologies, except in the case of CIFAR-10 for β values of 0.3 and 0.5. The values in the Table 1 for Eco-FL exceeding 1500 Wh (i.e., 15 Wh per 100 clients) are because, even when excluding clients with low remaining charge from the selection, the selected ones evidently contribute to a consumption higher than the remaining charge, along with the consumption of idle devices. In any case, this only happens in the final rounds because as clients exceed the threshold of 15 Wh, they can no longer be selected. Fig. 3 shows the values from Table 1. As observed, the energy consumption decreases for Eco-FL as the β value increases. The only exception, albeit minimal, occurs for the CINIC-10 dataset, where transitioning from $\beta = 0.3$ to $\beta = 0.5$ shows a slight increase in this value. This behavior is not always true for the other methods. In Fig. 4 a comparison of energy consumption per client across different methods, using $\beta = 0.3$ and varying datasets is reported. Note that Eco-FL is the method that requires the least consumption per client, while FedAvg is the one that requires the most. Eco-FL requires less energy consumption per client than FedEntropy in many cases, although the difference with the latter method is minor compared to other methods. Fig. 4 highlights that energy consumption is distributed among clients, without cases where some clients consume much more than others. This is because, overall, the clients are all selected to participate in the training.

In Figs. 5 and 6, values of energy consumption and accuracy during training for each method are shown, varying with β and dataset. In the various cases analyzed, it often happens that our method is the second to stop. This can be due either to a limit of energy available to all clients or to the target accuracy reached. Actually, looking at the accuracy graphs, we realize that the maximum consumption limit reached by the clients occurs only in the case of the CIFAR-100 dataset, where at the end of the FL process with our method, only a few clients are selected, leading to a decay in performance in the final rounds, as also verified in [30]. For the CIFAR-100 dataset, the 40% accuracy threshold is never reached before the end of the last round; therefore, Eco-FL terminates because the devices have exhausted their available energies, while the other methods continue until the allowed number of rounds is reached. However, there has been no improvement

Table 2

Total Wh consumed with homogeneous and heterogeneous initial energy availability for Eco-FL.

Dataset	β	Homogeneous	Heterogeneous
CIFAR-10	0.1	227	125
	0.3	200	109
	0.5	194	120
	1.0	91	123
CIFAR-100	0.1	1583	1589
	0.3	1546	1560
	0.5	1536	1579
	1.0	1524	1528
CINIC-10	0.1	1471	938
	0.3	637	394
	0.5	647	323
	1.0	550	254

in the accuracies of these methods, despite attempting to train the model for many more rounds. For CIFAR-10 and CINIC-10, instead, the process terminates due to reaching the fixed threshold value of 40%. In these cases, we observe that FedProx is the method with a slower rise in accuracy, which implies a greater number of rounds and consequently, a higher consumption of energy resources. Eco-FL, on the other hand, takes approximately the same number of rounds as FedAvg and FedEntropy, but with lower energy consumption. To demonstrate how device batteries drain during the FL process, we have depicted in Fig. 7 the drainage in the case of $\alpha = 0.5$ and $\beta = 0.5$ for each of the three datasets. During training, all devices are selected and contribute to the global model. The discharge is fairly uniform across all devices, without highlighting any particular spikes. In the case of CIFAR-10, due to the longer training process resulting from the more complex task of distinguishing between 20 classes instead of 10, some devices exhaust their battery towards the end of the training process. Hence, overall, the clients are all selected to participate in the training and there is not bias linked to the selection of clients.

5.1.2. Devices with heterogeneous initial energy availability

In this section, we share our experimental results using Eco-FL and heterogeneous starting available energies for all clients and the parameter α set to 0.5 in the optimization problem. Results for each of the three datasets, varying β are reported in Table 2, compared to those obtained in the previous homogeneous case. In the case of heterogeneous initial energies, Eco-FL is capable of reducing energy consumption compared to the homogeneous case, for CIFAR-10 and CINIC-10 datasets.

Comparing the results obtained in the heterogeneous case with those of the homogeneous case, globally Eco-FL consumes less in the case of heterogeneous devices, which is the most realistic scenario. In particular, for CIFAR-10 and CINIC-10 datasets, the costs are almost halved in all cases except for CIFAR-10 with $\beta = 1.0$. Of course, the higher the beta, the less non-IID the Dirichlet distribution is. Therefore, we expect Eco-FL's consumption to decrease as β increases, as the model achieves the same performance in fewer rounds and therefore consumes less. This is particularly evident in the case of CINIC-10. In the case of CIFAR-100, as in the homogeneous case, all clients are utilized to their maximum capacity. Therefore, the excess above 1500 Wh is due to the actual consumption during the last round in which the client is selected because it has residual energy greater than the threshold but then actually consumes more than it has available. In the subsequent round, each client in this situation can no longer be selected. The difference between the homogeneous and heterogeneous cases is therefore minimal. Note that in the heterogeneous case, the performance of FedAvg, FedProx, and FedEntropy remains unchanged compared to the homogeneous case, as these methods do not consider energy levels in the selection criteria.

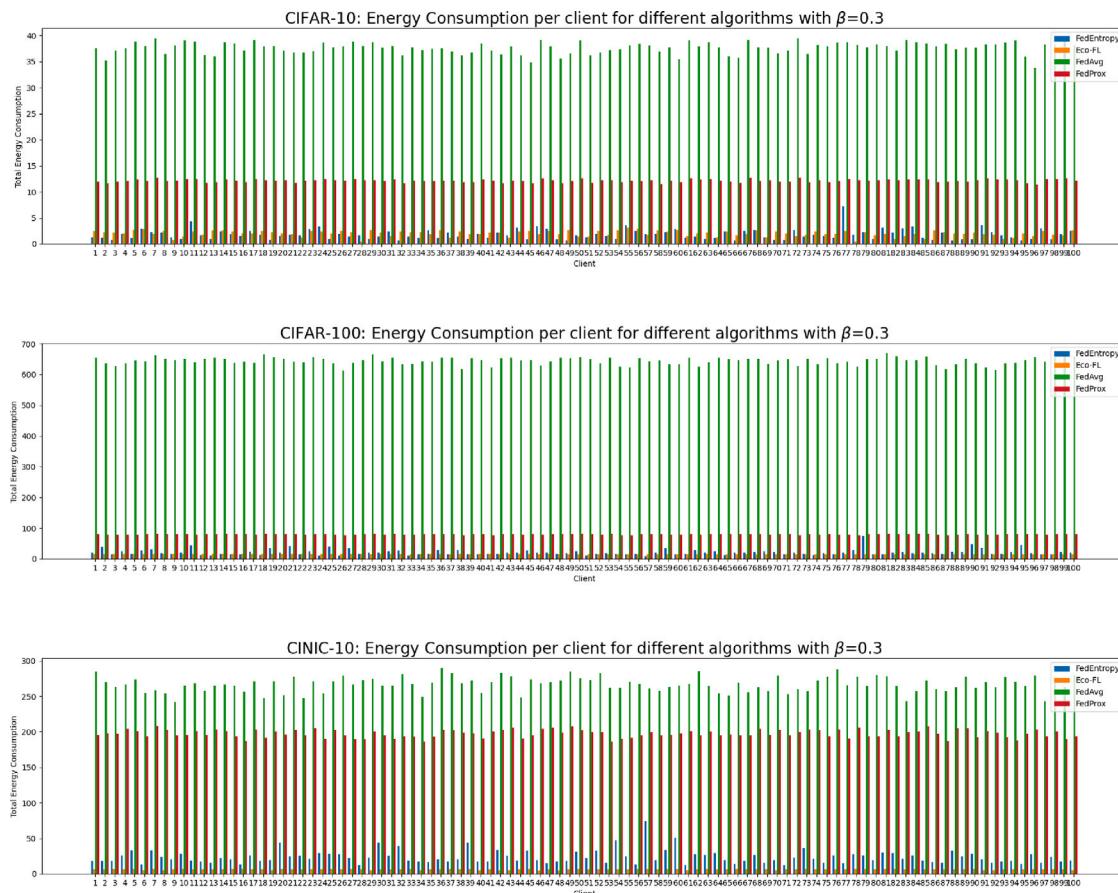


Fig. 4. Energy consumption per client for various algorithms, with $\beta = 0.3$ and varying datasets (CIFAR-10, CIFAR-100 and CINIC-10). Each bar represents the total energy consumption per client for a single algorithm. The bars are organized in groups corresponding to client indices, with each group containing bars for the four examined algorithms: FedEntropy, Eco-FL, FedAvg, and FedProx.

To illustrate the battery drain on devices during the FL process, we have showcased in Fig. 8 the depletion occurring under the conditions of $\alpha = 0.5$ and $\beta = 0.5$ across all three datasets. In this scenario, the selection of devices is not uniform, as in the homogeneous case, but rather more varied due to the diverse initial energy levels among the devices. Some devices are selected only in the later stages of training, after other devices have already depleted their available energy. Given the increased number of rounds required in the heterogeneous case, it is evident that many more devices deplete their energy reserves compared to the homogeneous case, as they are also utilized more extensively.

5.1.3. Variation of the parameter α in the objective function

We chose to test how the variation of the parameter α in the objective function affects the consumption of Eco-FL. Varying α means changing the weight associated with energy and entropy within the objective function. As α increases, more weight is given to energy and less to entropy. In the case of $\alpha = 1.0$, only energy is used to maximize the objective function. Results are reported in Table 3 and tests are conducted in the heterogeneous scenario. The dataset that results in the highest energy consumption is CIFAR-100 for each value of α . It is worth noting a decrease in energy consumption for this dataset in the case of $\beta = 1.0$ and $\alpha = 1.0$. For CIFAR-10 and CINIC-10, on average, energy consumption increases as α increases. This highlights the importance of partially considering entropy in the selection criterion, as both performance and the number of rounds required to achieve it depend on the data and how it is distributed among clients. Looking at the case $\alpha = 0.3$, there is an increase in consumed energy compared to the other cases, especially for CIFAR-100 with $\beta = 0.3$ and $\beta = 0.5$, and CINIC-10 with $\beta = 0.1$. This is because

using $\alpha = 0.3$ energy is not considered as much. In Table 4, accuracies at the last round with heterogeneous initial energy availability, varying the parameter α in the objective function, are reported, confirming that accuracies are almost unchanged when varying the α parameter. So, to maintain accuracies at the same level without excessive energy consumption, it is preferable to set a sufficiently large value for α , while still partially retaining the entropy information, which is essential for achieving rapid convergence and consequently lessening the energy impact.

5.2. Variation in the number of clients

We conducted tests varying the number of clients participating in FL. Specifically, we added scenarios with client counts set to 75 and 125. We fixed $\alpha = 0.5$ and conducted experiments varying the β parameter to divide datasets among clients. This allowed us to evaluate the performance of Eco-FL in different environments. The charts in Fig. 9 show the variation of energy consumption in dependence on the number of clients, varying β and the dataset, with $\alpha = 0.5$ fixed. Overall, CIFAR-10 exhibits low energy consumption across all β values, while CIFAR-100 shows high energy consumption regardless of β . CINIC-10 displays an intermediate trend, with energy consumption decreasing as β increases. Increasing the number of clients does not significantly alter the energy consumption pattern among the different β values. Instead, energy consumption varies significantly between datasets and is partially influenced by the data distribution.

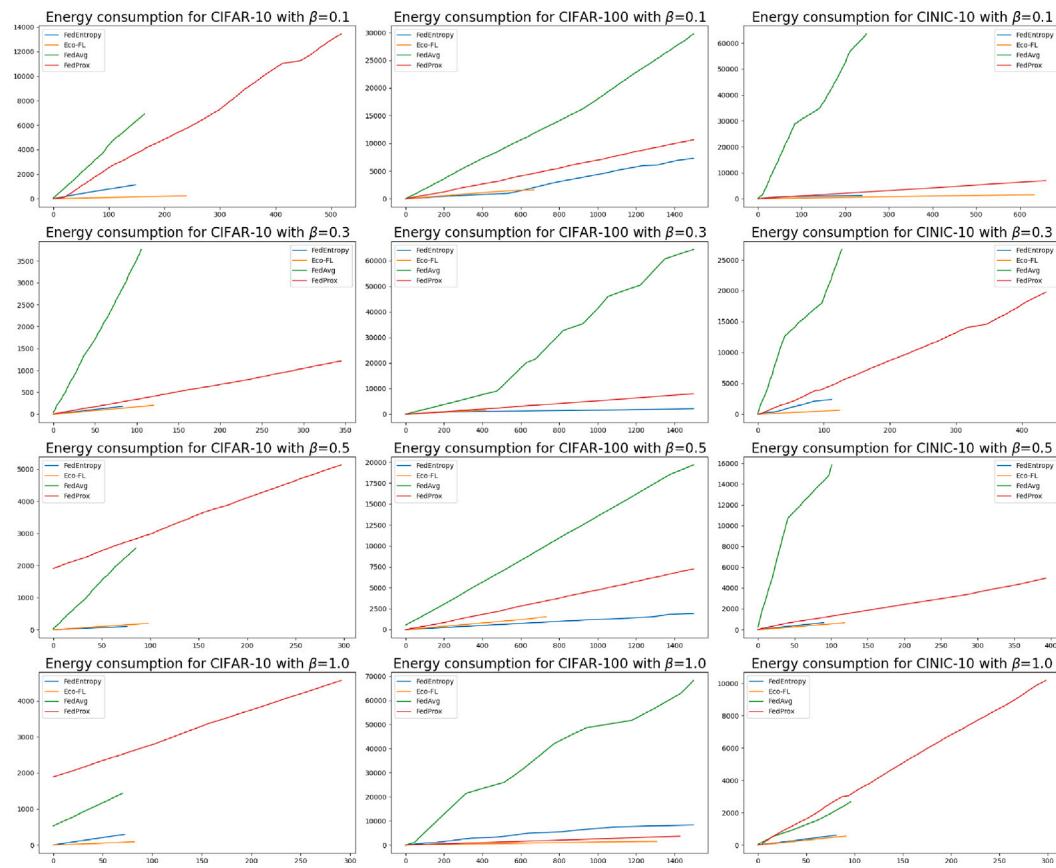


Fig. 5. Comparison of energy consumption in the homogeneous case, among the 4 methods (FedEntropy, Eco-FL, FedAvg, and FedProx), varying β values across the three selected datasets.

Table 3

The total Wh consumed with heterogeneous initial energy availability per Eco-FL, varying the parameter α in the objective function.

α	β	CIFAR-10	CIFAR-100	CINIC-10
0.3	0.1	113	1092	1650
	0.3	115	1631	407
	0.5	113	1625	383
	1.0	108	1499	311
0.5	0.1	125	1589	938
	0.3	109	1560	394
	0.5	120	1579	323
	1.0	123	1528	254
0.7	0.1	160	1504	918
	0.3	113	1526	584
	0.5	130	1535	643
	1.0	157	1519	575
1.0	0.1	376	1501	1197
	0.3	142	1519	704
	0.5	109	1515	618
	1.0	95	1001	537

5.3. Optimization problem and model weights aggregation costs

To compare the total consumptions observed across the 3 datasets with the consumptions of the model weight aggregation and optimization problem operations, the latter were calculated and reported in Fig. 10. Specifically, the figure depicts the scenario of 100 clients with α and β both set to 0.5. The consumption related to these two operations is low when compared to the total consumption. Additionally, we observe that the dataset that consumes the most to perform both of these operations is CIFAR-100. Comparing the consumption with

Table 4

Accuracy (%) at the last round with heterogeneous initial energy availability per Eco-FL, varying the parameter α in the objective function.

α	β	CIFAR-10	CIFAR-100	CINIC-10
0.3	0.1	38.83	32.13	33.60
	0.3	38.40	28.06	38.42
	0.5	36.47	30.45	36.52
	1.0	42.08	30.96	39.72
0.5	0.1	40.29	21.58	39.55
	0.3	38.69	30.05	39.86
	0.5	42.63	32.15	38.84
	1.0	41.47	28.87	39.57
0.7	0.1	32.69	28.44	32.89
	0.3	36.27	30.91	37.88
	0.5	38.10	31.03	40.87
	1.0	41.55	32.55	40.59
1.0	0.1	40.36	25.54	36.35
	0.3	40.08	30.31	39.30
	0.5	42.38	24.36	39.36
	1.0	41.40	39.04	40.14

the total consumption (Fig. 9(b)), it emerges that these operations overall consume less than 10% of the total energy. This underscores the efficiency and minimal impact of these operations on the overall energy consumption within the system, underlying that the consumption related to the optimization problem is comparable to that of the weights aggregation.

5.4. Additional metrics in eco-FL evaluation

In FL assessing performance requires a comprehensive understanding of various metrics, such as throughput, latency and convergence

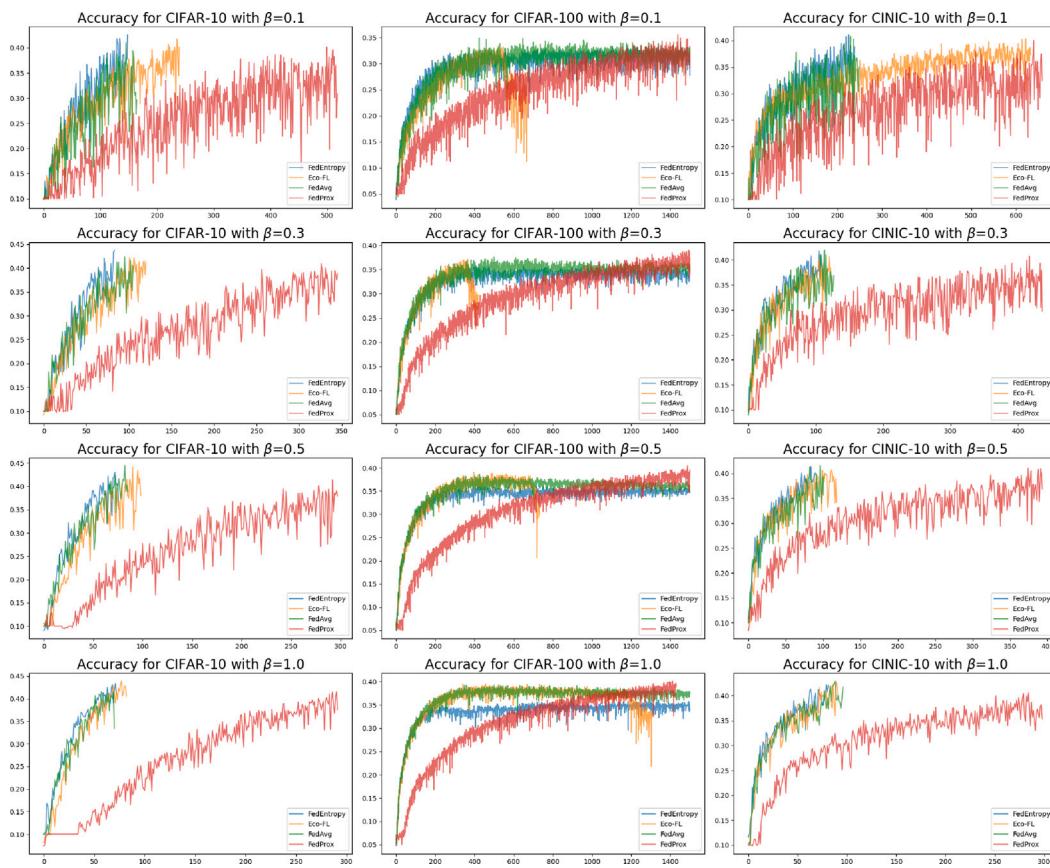


Fig. 6. Comparison of accuracy in the homogeneous case, among the 4 methods (FedEntropy, Eco-FL, FedAvg, and FedProx), varying β values across the three selected datasets. Each line represents a client.

time. To illustrate these aspects, all the following results are in the scenario of heterogeneous initial energy, with α set to 0.5 and encompassing a total of 100 clients.

5.4.1. Convergence time

We examined how different β settings influence the time required to train the model. As depicted in Fig. 11, for both CIFAR-10 and CINIC-10, varying the number of clients, the maximum time, as expected, is for $\beta = 0.1$. Increasing the β values, the times decrease, especially in the case of CINIC-10, where the difference between different values of β is more pronounced. CIFAR-100 requires more time compared to other datasets and achieves, especially for $\beta = 1.0$, an impressive high consumption, due to more training rounds required to stop the training process.

5.4.2. Latency

Fixing the number of clients to 125, we present the latency incurred by local training across other clients, as well as the time required for model aggregation post each training round and the duration for solving the optimization problem in each round, varying β and the dataset (Figs. 12, 13, 14). The latency attributed to the training of clients, which incurs longer processing times, peaks at approximately 2 seconds when utilizing the CIFAR-10 dataset. This latency tends to be higher than with the other two datasets, wherein maximum latency periods consistently remain below a second. In general, for CIFAR-10 and CINIC-10, as the value of β in the Dirichlet distribution increases, latency decreases. When considering the waiting time for the resolution of the optimization problem for client selection, the highest peaks in time are observed for CIFAR-100. These peaks are scattered between approximately 0.2 and 0.4 s. Conversely, with the other two datasets,

the times consistently remain below one-tenth of a second. The weight aggregation times for each round generally remain below one-tenth of a second, except for a peak around round 260 in CIFAR-100, which requires more than 0.4 s. This isolated peak could be attributed to a decrease in the connection speed between the clients and the server. Overall, the operation that consumes the most time in terms of latency for the clients is waiting for the completion of training for all selected clients.

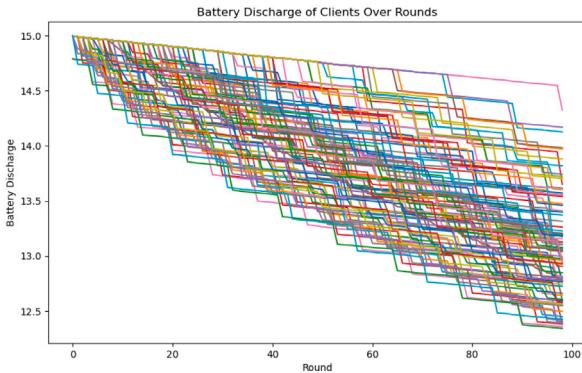
5.4.3. Throughput

At the beginning of FL, the throughput values for data loading of 100 clients are: 42 GB/s for CIFAR-10, 43 GB/s for CIFAR-100, and 140 GB/s for CINIC-10. These high throughput values for data loading ensure fast initial setup. Transmitting the data related to the energy consumed at each round requires a throughput of 8×10^{-8} GB/s for CIFAR-10, 7×10^{-8} GB/s for CIFAR-100 and 3×10^{-8} GB/s for CINIC-10, ensuring that the communication network is not significantly busied by these additional data transmissions, leaving more bandwidth available for other operations.

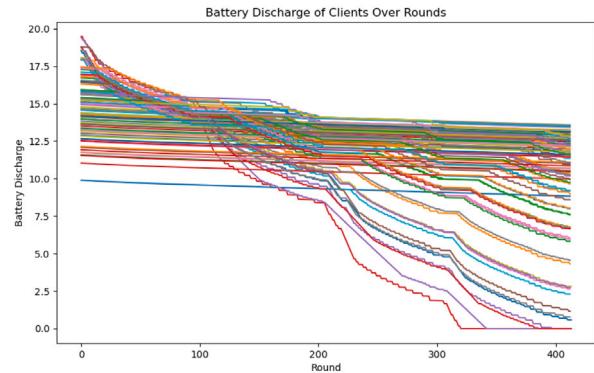
5.5. Device failures in Eco-FL

Devices participating in FL can encounter various challenges, such as intermittent connectivity issues or sudden battery discharge, which can disrupt the training process. To address these concerns, we conducted tests simulating scenarios where devices experience temporary problems during FL rounds.

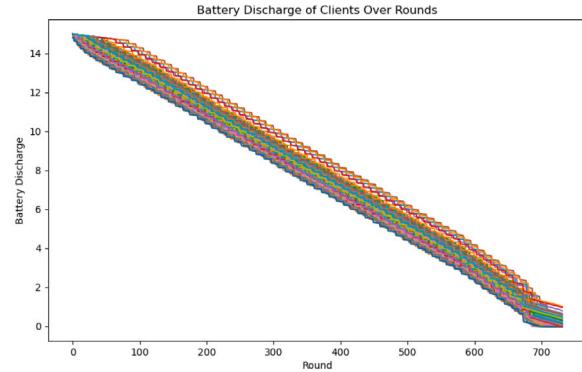
In our testing procedure, we simulated a scenario where, in each round, there is a probability of $\frac{1}{10}$ that some selected devices will encounter temporary issues such as network disruptions or connection



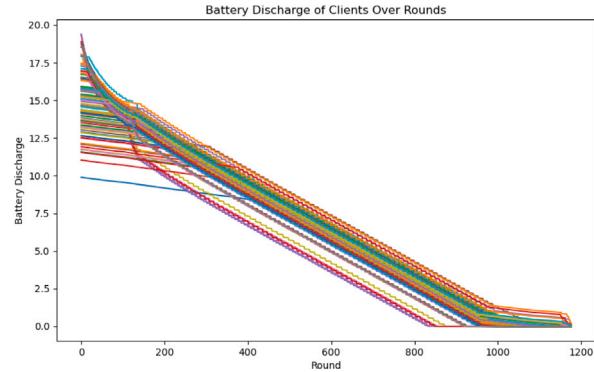
(a) CIFAR-10



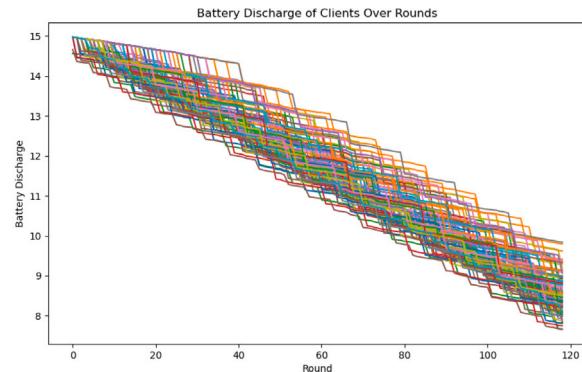
(a) CIFAR-10



(b) CIFAR-100



(b) CIFAR-100



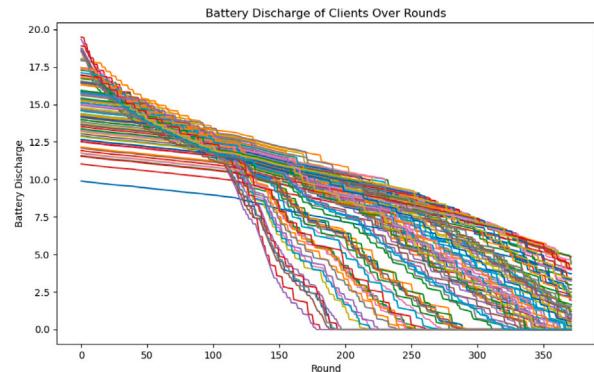
(c) CINIC-10

Fig. 7. Device batteries drain in the homogeneous case using $\alpha = 0.5$ and $\beta = 0.5$ for each of the three datasets. Each line represents a client.

fluctuations. The number of affected devices is randomly determined. These devices are temporarily excluded from the current round but are reinstated as selectable in the next round.

Additionally, in each round, with a probability of $\frac{1}{10}$ among the selected, devices are randomly chosen to be incapable of training the model due to sudden battery drain, and subsequently excluded from the training phase and the selectable set of devices in the subsequent rounds.

These two types of issues can occur simultaneously in the same round. Fig. 15 demonstrates the client selection process of Eco-FL during training as β varies, with $\alpha = 0.5$ in the heterogeneous case involving 100 clients. As shown, numerous devices become unavailable during the training process and are subsequently excluded from the



(c) CINIC-10

Fig. 8. Device batteries drain in the heterogeneous case using $\alpha = 0.5$ and $\beta = 0.5$ for each of the three datasets. Each line represents a client.

current round's training. This results in a noticeable fluctuation in the number of clients contributing to the global model.

Using fewer clients results in the utilization of less dataset information, which is reflected in slightly lower accuracy in all cases (Table 5). In these cases, the procedure stops due to the lack of available clients in the client selection, caused by sudden discharges or the depletion of the batteries of the devices used. In particular, the performance of CIFAR-100 declines significantly. With 20 classes to distinguish, compared to 10 in the other two datasets, a greater number of rounds is required to achieve the same accuracy as the other two datasets. However, this is not possible in such a situation, due to the unavailability of devices.

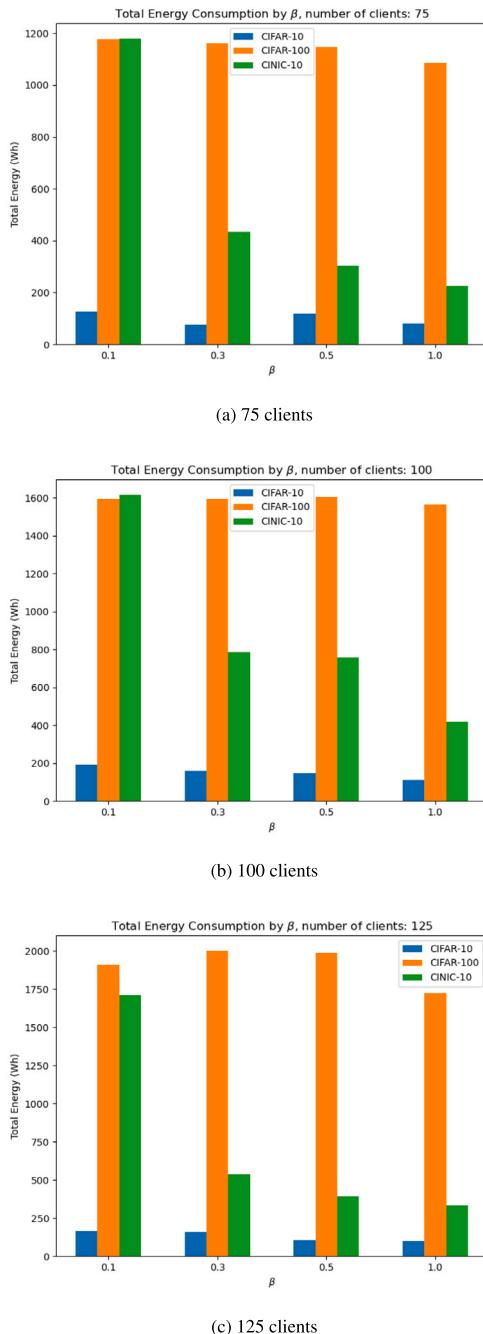


Fig. 9. Variation of energy consumption depending on the number of clients, varying β and the dataset, with $\alpha = 0.5$ fixed.

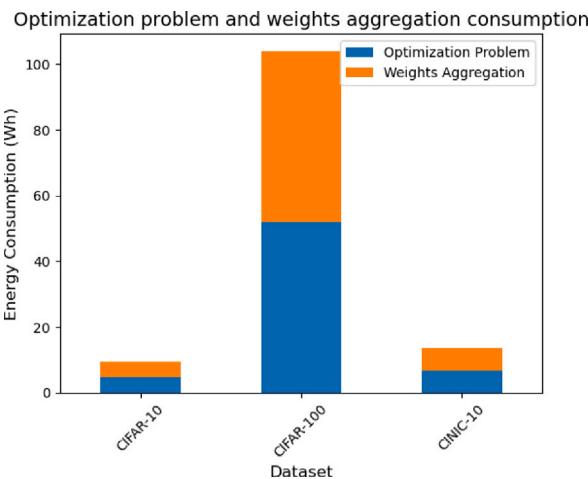


Fig. 10. Consumptions of the model weights aggregation and optimization problem operations. Each bar represents the overall total consumed energy, while the different colored sections within each bar indicate the operations.

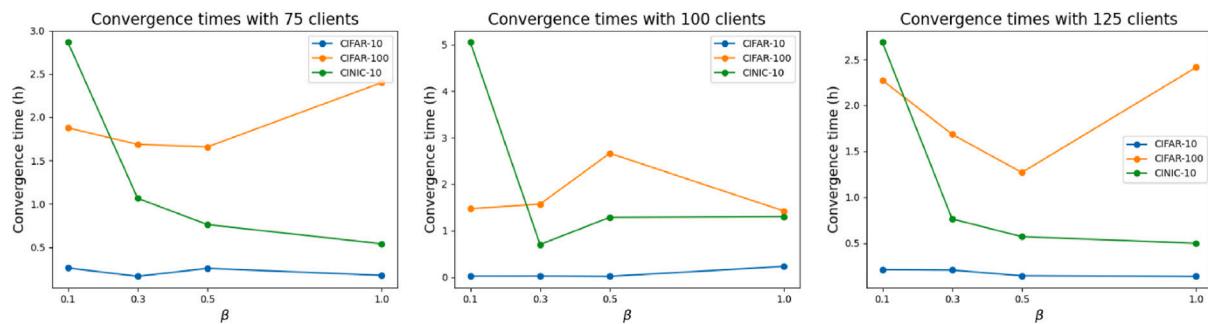


Fig. 11. Convergence times, varying the number of clients, β and the dataset, with $\alpha = 0.5$ fixed.

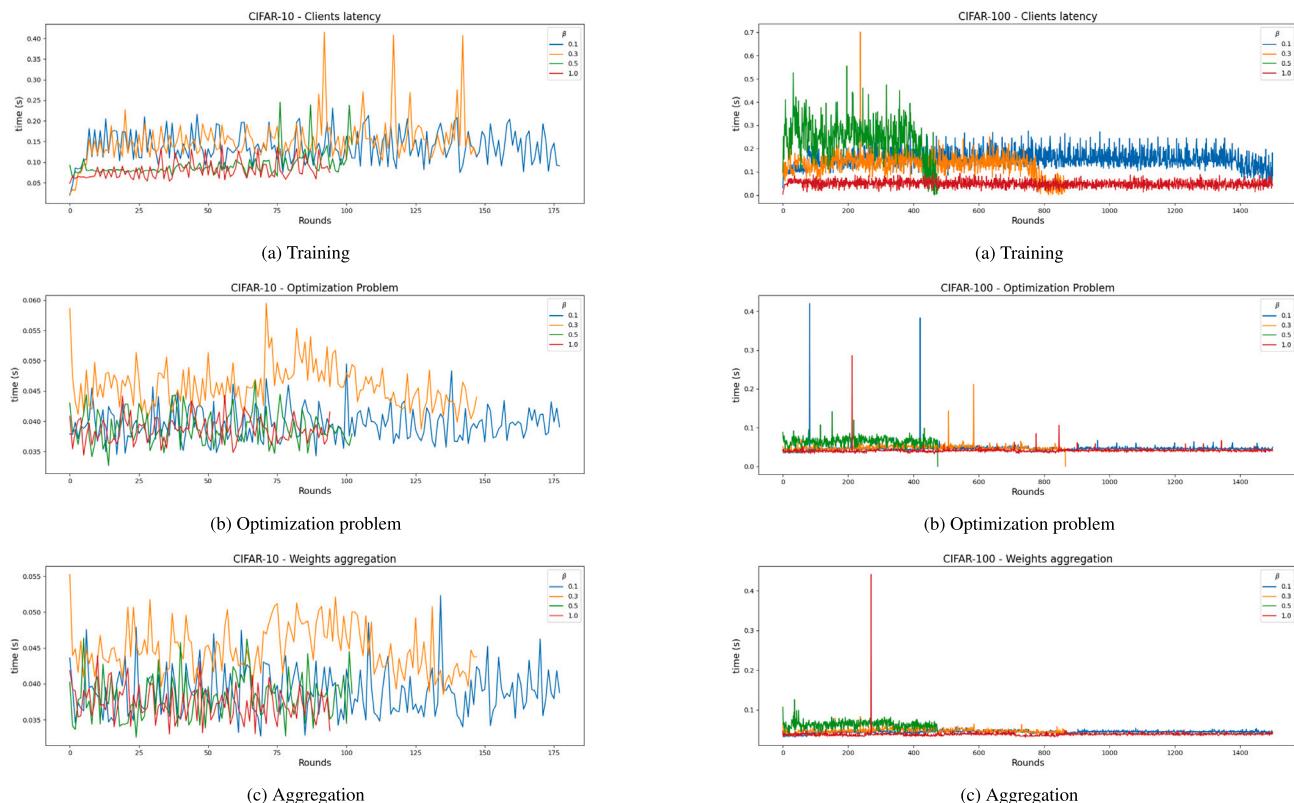
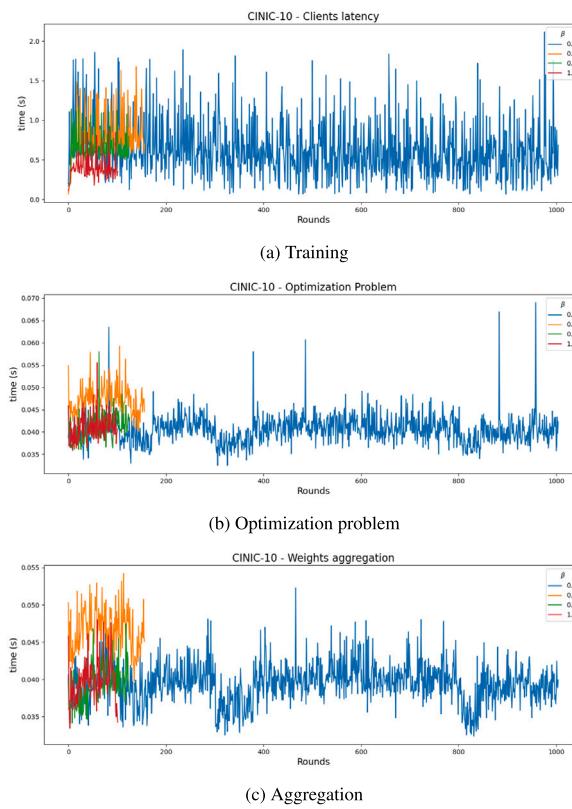


Fig. 12. CIFAR-10 varying β .

Fig. 13. CIFAR-100 varying β .

Fig. 14. CINIC-10 varying β .**Table 5**

Accuracy (%) and energy consumption (Wh) results when device failure is allowed, for different values of β and datasets in the heterogeneous case.

Dataset	β	Accuracy	Energy consumption
CIFAR-10	0.1	37.82	680
	0.3	37.62	262
	0.5	42.75	322
	1.0	38.53	196
CIFAR-100	0.1	13.68	643
	0.3	16.42	657
	0.5	18.58	757
	1.0	16.42	772
CINIC-10	0.1	29.65	1749
	0.3	35.81	570
	0.5	41.53	1112
	1.0	35.81	819

6. Conclusions

To address the challenge of energy consumption in FL, this work provides a solution through a client selection methodology. By evaluating FL algorithms across different datasets and energy scenarios, we have demonstrated the efficacy of our proposed methodology in optimizing client selection based on data entropy and available energy. Our research contributes to enhancing the sustainability of FL, especially in resource-constrained environments such as edge cloud-computing. Our findings reveal that selecting clients with higher residual energy can significantly enhance the utilization of resources and improve the overall performance of the FL process. We observed that considering data entropy is also crucial for achieving the optimal solution. In addition, our experiments highlight the importance of considering both homogeneous and heterogeneous energy distributions among clients, as these factors significantly influence the energy consumption and

efficiency of FL. We also tested Eco-FL in a more realistic scenario where devices may experience connection issues or sudden battery drains, and with varying total numbers of clients. By considering factors such as data entropy and available energy, our approach contributes to enhancing the sustainability and efficiency of Federated Learning, particularly in resource-constrained environments.

CRediT authorship contribution statement

Martina Savoia: Writing – original draft, Visualization, Formal analysis, Data curation, Conceptualization. **Edoardo Prezioso:** Writing – original draft, Validation, Investigation. **Valeria Mele:** Writing – original draft, Supervision, Data curation. **Francesco Piccialli:** Writing – review & editing, Validation, Supervision, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

- PNRR project FAIR - Future AI Research (PE00000013), Spoke 3, under the NRRP MUR program funded by the NextGenerationEU.
- PNRR Centro Nazionale HPC, Big Data e Quantum Computing, (CN_00000013)(CUP: E63C22000980007), under the NRRP MUR program funded by the NextGenerationEU.

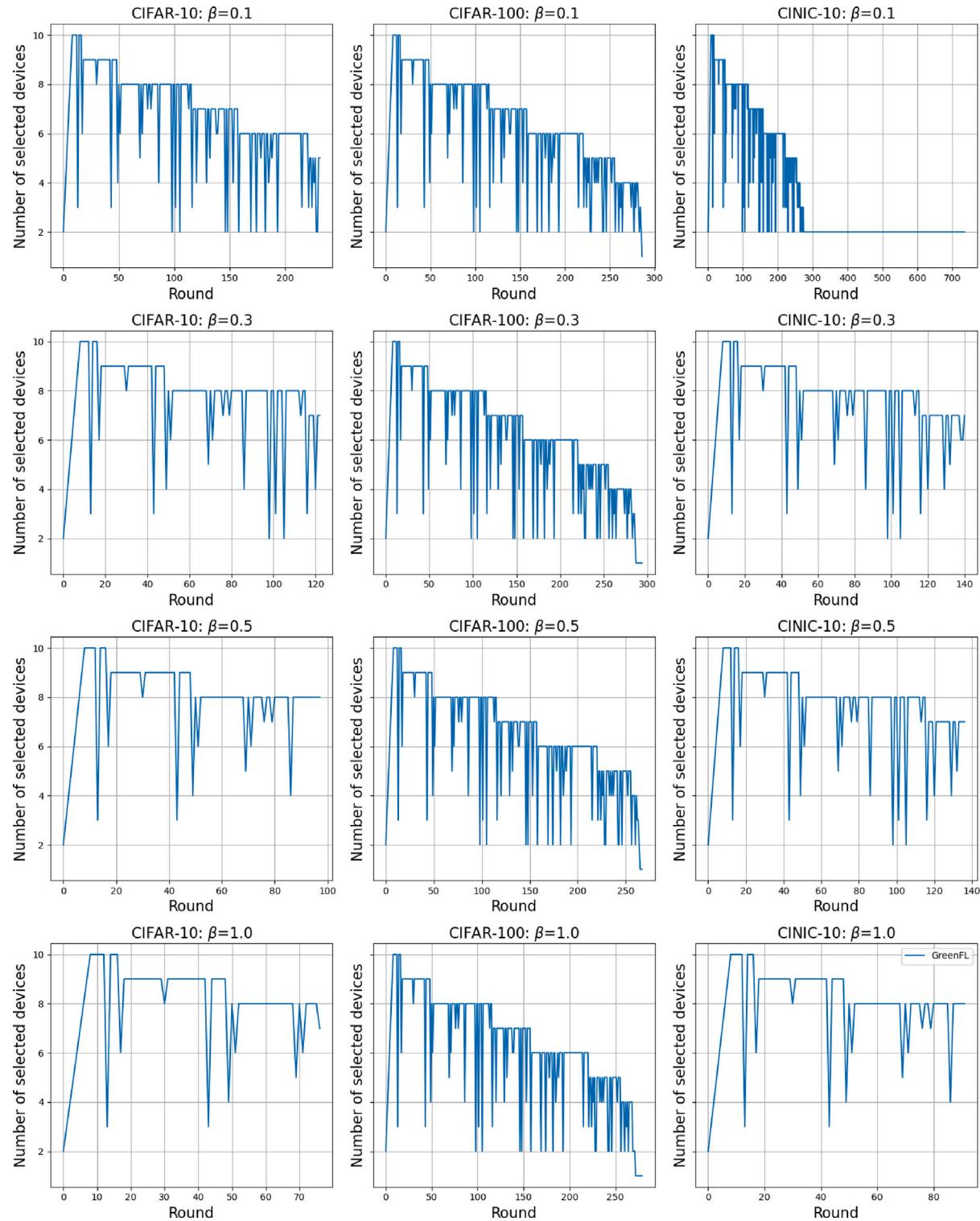


Fig. 15. Number of selected devices per round when device failure is allowed, using $\alpha = 0.5$ and 100 clients in the heterogeneous case. There is a notable fluctuation in the number of clients contributing to the global model.

- G.A.N.D.A.L.F. - Gan Approaches for Non-iID Aiding Learning in Federations, CUP: E53D23008290006, PNRR - Missione 4 “Istruzione e Ricerca” - Componente C2 Investimento 1.1 “Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN)”.

References

- [1] T.L. Duc, R.G. Leiva, P. Casari, P.O. Östberg, Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey, *ACM Comput. Surv.* 52 (5) (2019) <http://dx.doi.org/10.1145/3341145>.
- [2] K. Cao, S. Hu, Y. Shi, A.W. Colombo, S. Karnouskos, X. Li, A survey on edge and edge-cloud computing assisted cyber-physical systems, *IEEE Trans. Ind. Inform.* 17 (11) (2021) 7806–7819.
- [3] J. Konečný, B. McMahan, D. Ramage, Federated optimization: Distributed optimization beyond the datacenter, 2015, arXiv preprint [arXiv:1511.03575](https://arxiv.org/abs/1511.03575).
- [4] L. Li, Y. Fan, M. Tse, K.Y. Lin, A review of applications in federated learning, *Comput. Ind. Eng.* 149 (2020) 106854.
- [5] H. Zhu, J. Xu, S. Liu, Y. Jin, Federated learning on non-IID data: A survey, *Neurocomputing* 465 (2021) 371–390.
- [6] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1273–1282.
- [7] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, *Proc. Mach. Learn. Syst.* 2 (2020) 429–450.
- [8] G. Laccetti, M. Lapeagna, V. Mele, D. Romano, A. Murli, A double adaptive algorithm for multidimensional integration on multicore based HPC systems, *Int. J. Parallel Program.* 40 (2012) 397–409.
- [9] G. Laccetti, M. Lapeagna, V. Mele, D. Romano, A study on adaptive algorithms for numerical quadrature on heterogeneous gpu and multicore based systems, in: *Parallel Processing and Applied Mathematics: 10th International Conference, PPAM 2013, Warsaw, Poland, September 8–11, 2013, Revised Selected Papers, Part I* 10, Springer, 2014, pp. 704–713.
- [10] V. Boccia, L. Carracciolo, G. Laccetti, M. Lapeagna, V. Mele, HADAB: enabling fault tolerance in parallel applications running in distributed environments, in: *Parallel Processing and Applied Mathematics: 9th International Conference, PPAM 2011, Torun, Poland, September 11–14, 2011. Revised Selected Papers, Part I* 9, Springer, 2012, pp. 700–709.
- [11] J. Xu, B.S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, *J. Healthc. Inform. Res.* 5 (2021) 1–19.
- [12] P. Sharma, F.E. Shamout, D.A. Clifton, Preserving patient privacy while training a predictive model of in-hospital mortality, 2019, arXiv preprint [arXiv:1912.00354](https://arxiv.org/abs/1912.00354).
- [13] G. Long, Y. Tan, J. Jiang, C. Zhang, Federated learning for open banking, in: *Federated Learning: Privacy and Incentive*, Springer, 2020, pp. 240–254.
- [14] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, 2018, arXiv preprint [arXiv:1811.03604](https://arxiv.org/abs/1811.03604).
- [15] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Federated learning for internet of things: A comprehensive survey, *IEEE Commun. Surv. Tutor.* 23 (3) (2021) 1622–1658.
- [16] X. Zhou, W. Huang, W. Liang, Z. Yan, J. Ma, Y. Pan, I. Kevin, K. Wang, Federated distillation and blockchain empowered secure knowledge sharing for Internet of medical Things, *Inform. Sci.* 662 (2024) 120217.
- [17] L. Yuan, Z. Wang, L. Sun, S.Y. Philip, C.G. Brinton, Decentralized federated learning: A survey and perspective, *IEEE Internet Things J.* (2024).
- [18] X. Zhou, W. Liang, I. Kevin, K. Wang, Z. Yan, L.T. Yang, W. Wei, J. Ma, Q. Jin, Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems, *IEEE Wirel. Commun.* 30 (2) (2023) 82–89.
- [19] X. Zhou, Q. Yang, X. Zheng, W. Liang, I. Kevin, K. Wang, J. Ma, Y. Pan, Q. Jin, Personalized federation learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse, *IEEE J. Sel. Areas Commun.* (2024).
- [20] X. Zhou, X. Zheng, X. Cui, J. Shi, W. Liang, Z. Yan, L.T. Yang, S. Shimizu, I. Kevin, K. Wang, Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks, *IEEE J. Sel. Areas Commun.* (2023).
- [21] S.P. Ramu, P. Boopalan, Q.V. Pham, P.K.R. Maddikunta, T. Huynh-The, M. Alazab, T.T. Nguyen, T.R. Gadekallu, Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions, *Sustainable Cities Soc.* 79 (2022) 103663.
- [22] X. Zhou, Q. Yang, Q. Liu, W. Liang, K. Wang, Z. Liu, J. Ma, Q. Jin, Spatial-temporal federated transfer learning with multi-sensor data fusion for cooperative positioning, *Inf. Fusion* 105 (2024) 102182.
- [23] T. Qi, Y. Zhan, P. Li, J. Guo, Y. Xia, Hwamei: A learning-based synchronization scheme for hierarchical federated learning, in: *2023 IEEE 43rd International Conference on Distributed Computing Systems, ICDCS, IEEE*, 2023, pp. 534–544.
- [24] X. Zhou, W. Liang, A. Kawai, K. Fueda, J. She, I. Kevin, K. Wang, Adaptive segmentation enhanced asynchronous federated learning for sustainable intelligent transportation systems, *IEEE Trans. Intell. Transp. Syst.* (2024).
- [25] Z. Xu, L. Li, W. Zou, Exploring federated learning on battery-powered devices, in: *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–6.
- [26] M. Kapsecker, D.N. Nugraha, C. Weinhuber, N. Lane, S.M. Jonas, Federated learning with swift: An extension of flower and performance evaluation, *SoftwareX* 24 (2023) 101533, <http://dx.doi.org/10.1016/j.softx.2023.101533>, URL <https://www.sciencedirect.com/science/article/pii/S2352711023002297>.
- [27] B. Guler, A. Yener, Sustainable federated learning, 2021, arXiv preprint [arXiv:2102.11274](https://arxiv.org/abs/2102.11274).
- [28] X. Zhou, X. Ye, I. Kevin, K. Wang, W. Liang, N.K.C. Nair, S. Shimizu, Z. Yan, Q. Jin, Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications, *IEEE Trans. Comput. Soc. Syst.* (2023).
- [29] R. Albelaihi, L. Yu, W.D. Craft, X. Sun, C. Wang, R. Gazda, Green federated learning via energy-aware client selection, in: *GLOBECOM 2022-2022 IEEE Global Communications Conference*, IEEE, 2022, pp. 13–18.
- [30] J. Xu, H. Wang, Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective, *IEEE Trans. Wireless Commun.* 20 (2) (2020) 1188–1200.
- [31] R. Albelaihi, X. Sun, W.D. Craft, L. Yu, C. Wang, Adaptive participant selection in heterogeneous federated learning, in: *2021 IEEE Global Communications Conference, GLOBECOM, IEEE*, 2021, pp. 1–6.
- [32] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: *ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE*, 2019, pp. 1–7.
- [33] Z. Ling, Z. Yue, J. Xia, M. Hu, T. Wang, M. Chen, FedEntropy: Efficient device grouping for federated learning using maximum entropy judgment, 2022, arXiv preprint [arXiv:2205.12038](https://arxiv.org/abs/2205.12038).
- [34] M. Ribero, H. Vikalo, Reducing communication in federated learning via efficient client sampling, *Pattern Recognit.* 148 (2024) 110122.
- [35] C. Wang, X. Wei, P. Zhou, Optimize scheduling of federated learning on battery-powered mobile devices, in: *2020 IEEE International Parallel and Distributed Processing Symposium, IPDPS*, 2020, pp. 212–221.
- [36] A. Yousefpour, S. Guo, A. Shenoy, S. Ghosh, P. Stock, K. Maeng, S.W. Krüger, M. Rabbat, C.J. Wu, I. Mironov, Green federated learning, 2023, arXiv preprint [arXiv:2303.14604](https://arxiv.org/abs/2303.14604).
- [37] M. Kim, W. Saad, M. Mozaffari, M. Debbah, Green, quantized federated learning over wireless networks: An energy-efficient design, *IEEE Trans. Wireless Commun.* (2023).
- [38] A. Salhi, R. Ngah, L. Audah, K.S. Kim, Q. Abdullah, Y.M. Al-Moliki, K.A. Aljaloud, H.N. Talib, Energy-efficient federated learning with resource allocation for green IoT edge intelligence in B5G, *IEEE Access* 11 (2023) 16353–16367.
- [39] A. Salhi, R. Ngah, L. Audah, K.S. Kim, Q. Abdullah, Y.M. Al-Moliki, K.A. Aljaloud, H.N. Talib, Energy-efficient federated learning with resource allocation for green IoT edge intelligence in B5G, *IEEE Access* 11 (2023) 16353–16367.
- [40] P. Li, X. Huang, M. Pan, R. Yu, FedGreen: Federated learning with fine-grained gradient compression for green mobile edge computing, in: *2021 IEEE Global Communications Conference, GLOBECOM, IEEE*, 2021, pp. 1–6.
- [41] R. Deng, R. Lu, C. Lai, T.H. Luan, H. Liang, Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption, *IEEE Internet Things J.* 3 (6) (2016) 1171–1181.
- [42] S. Zheng, X. Wang, L. Duan, Adaptive federated learning via entropy approach, 2023, arXiv preprint [arXiv:2303.14966](https://arxiv.org/abs/2303.14966).
- [43] Y. Sun, S. Zhou, D. Gündüz, Energy-aware analog aggregation for federated learning with redundant data, in: *ICC 2020-2020 IEEE International Conference on Communications, ICC, IEEE*, 2020, pp. 1–7.
- [44] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K.H. Li, T. Parcollet, P.P.B. de Gusmão, et al., Flower: A friendly federated learning framework, 2022.
- [45] G.B. Barone, D. Bottalico, L. Carracciolo, A. Doria, D. Michelino, S. Pardi, G. Russo, G. Sabella, B. Spisso, Designing and implementing a high-performance computing heterogeneous cluster, in: *2022 International Conference on Electrical, Computer and Energy Technologies, ICECET*, 2022, pp. 1–6, <http://dx.doi.org/10.1109/ICECET55527.2022.9872709>.
- [46] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Toronto, ON, Canada, 2009.
- [47] L.N. Darlow, E.J. Crowley, A. Antoniou, A.J. Storkey, Cinic-10 is not imagenet or cifar-10, 2018, arXiv preprint [arXiv:1810.03505](https://arxiv.org/abs/1810.03505).