# CEFL: Carbon-Efficient Federated Learning

Talha Mehboob
University of Massachusetts Amherst

Noman Bashir
Massachusetts Institute of Technology

Jesus Omana Iglesias
Telefonica Research

Michael Zink
University of Massachusetts Amherst

David Irwin
University of Massachusetts Amherst

## Abstract

Federated Learning (FL) distributes machine learning (ML) training across many edge devices to reduce data transfer overhead and protect data privacy. Since FL model training may span millions of devices and is thus resource-intensive, prior work has focused on improving its resource efficiency to optimize time-to-accuracy. However, prior work generally treats all resources the same, while, in practice, they may incur widely different costs, which instead motivates optimizing cost-to-accuracy. To address the problem, we design CEFL, which uses adaptive cost-aware client selection policies to optimize an arbitrary cost metric when training FL models. Our policies extend and combine prior work on utility-based client selection and critical learning periods by making them cost-aware. We demonstrate CEFL by designing carbon-efficient FL, where energy's carbon-intensity is the cost, and show that it i) reduces carbon emissions by 93% and reduces training time by 50% compared to random client selection and ii) reduces carbon emissions by 80%, while only increasing training time by 38%, compared to a state-of-the-art approach that optimizes training time.

***CCS Concepts:*** • **Hardware → Impact on the environment**; • **Computing methodologies → Machine learning**; • **General and reference → Metrics**.

***Keywords:*** resource efficiency, cost, carbon, sustainability

## 1 Introduction

Federated Learning (FL) is an increasingly popular machine learning (ML) paradigm that distributes model training across thousands-to-millions of edge devices (or clients) across the wide-area, such as smartphones and IoT devices [34]. FL increases efficiency and security relative to centralized ML because it processes large input data locally and only sends updated model parameters to the central site, eliminating high data transfer overheads and protecting user privacy i.e., by ensuring raw input data never leaves an edge device. In some cases, FL may also need to comply with data protection laws, such as GDPR [12]. Given the large number of FL clients, training typically only selects a fraction of clients (often randomly) in each round. These selected clients perform multiple training iterations, e.g., by applying stochastic gradient descent, on the data before sending a model update to the central site, aka the aggregator, for aggregation with the global ML model.

Importantly, FL also differs from centralized ML in that the raw input data often derives from sensors on edge devices, e.g., cameras, microphones, accelerometers, etc., rather than stored datasets, and thus FL clients may be exposed to data with widely different characteristics. That is, data is often not independent and identically distributed (i.i.d.) across client devices. As a result, the specific clients selected each round significantly affect the time a model takes to reach a target accuracy, i.e., its time-to-accuracy. For example, repeatedly selecting clients that lack certain classes of data may result in a model that cannot accurately identify these classes.

To mitigate the problem, FL often randomly selects different subsets of clients for training over many rounds. However, this random approach can significantly increase the time-to-accuracy, and require many rounds of training to ensure sufficient data coverage. Thus, the random approach may potentially waste substantial resources by continually training on data that contributes little to improving a model's marginal accuracy, i.e., by frequently selecting data that the model has already trained on. As a result, recent work focuses on intelligent participant selection that chooses clients each round based on the statistical "utility" of their data, which captures how much a client's data contributes to improving the global model's accuracy [1, 28]. This recent work shows that intelligent participant selection can significantly reduce both resource waste and a model's time-to-accuracy.

However, while reducing time-to-accuracy is indeed important, prior work generally does not consider that clients may incur widely different *costs* for local training. For example, local training consumes energy, and energy's monetary cost may differ substantially across clients if they are geographically distributed. Similarly, energy's carbon-intensity may also differ widely across clients, so local training carbon "cost" also differs. While FL clients individually often consume little energy and carbon, collectively, their energy and carbon cost can be significant. In particular, recent work suggests that smartphones in aggregate have a significantly higher carbon footprint than cloud datacenters [38]. The cost metric could also be more abstract, such as the percentage of remaining battery power required for a training round. In this case, selecting clients for training with near-empty batteries would incur a higher cost than those with near-full batteries, since training may cause these clients to run out of energy. For many scenarios, a model's cost-to-accuracy is just as, if not more, important than its time-to-accuracy.

To address the problem, we present CEFL, which optimizes FL's cost-to-accuracy. Specifically, we design adaptive cost-aware client selection policies for CEFL that consider clients' data utility and cost and the current model state to determine which and how many clients to select for each training round. CEFL's policies combine and extend recent work on both utility-based client selection [1, 28] and critical learning periods [3, 44] by making them cost-aware. *Utility-based client selection* focuses on selecting clients with data that will contribute most to improving the global model's marginal accuracy. This data generally includes classes that are less represented by previously selected clients. The *critical learning period* represents the initial rounds of training that bootstrap the model weights. This period is particularly important because the model weights' early values have a significant influence on time-to-accuracy. In particular, aggressively selecting clients early, rather than late, in the training process can significantly improve time-to-accuracy. CEFL extends utility-based client selection and the critical learning period to optimize cost instead of time-to-accuracy. That said, time-to-accuracy influences cost, since training ceases to incur any cost after reaching its target accuracy.

As we discuss, optimizing cost requires judiciously deciding *which* clients to select based on both their cost and data utility, as well as *how many* to select and *when*. Our hypothesis is that, by extending utility-based cost selection and critical learning periods to consider cost, CEFL can significantly lower the cost-to-accuracy compared to state-of-the-art cost-agnostic approaches and cost-aware approaches that apply these techniques in isolation. In evaluating our hypothesis, this paper makes the following contributions.

**Illustrative Cost Analysis.** We use client carbon emissions to illustrate the importance of cost-aware client selection in FL. Specifically, we analyze data to show that clients' carbon costs can differ widely for local training.

**Cost-aware Client Selection.** We design an adaptive cost-aware client selection policy for CEFL. In particular, we first extend both utility-based client selection and client selection during critical learning periods to make them cost-aware and then combine both approaches to further reduce cost.

**Implementation and Evaluation.** We implement CEFL and its adaptive cost-aware client selection policies in the Flower framework [4], and evaluate it across multiple real datasets under different distributions of data across clients. Our results show that CEFL i) reduces carbon emissions by 93% and reduces training time by 50% compared to random client selection and ii) reduces carbon emissions by 80%, while only increasing training time by 38%, compared to a state-of-the-art approach that optimizes training time.

## 2 Ingredients of Carbon-Efficient FL

Below, we provide an overview of federated learning (FL) (§2.1) and discuss related work on resource-efficient client selection (§2.2). We then introduce our notion of cost and potential cost metrics in FL (§2.3) and discuss the various factors that influence client selection (§2.4).

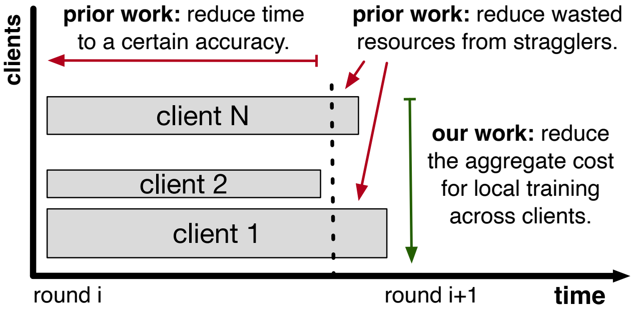

clients
prior work: reduce time to a certain accuracy.
prior work: reduce wasted resources from stragglers.
client N
client 2
client 1
our work: reduce the aggregate cost for local training across clients.
round i
round i+1
time

**Figure 1.** ***CEFL's notion of cost: Prior work focuses on enhancing system efficiency by reducing time-to-convergence and eliminating wasted computations from stragglers. CEFL focuses on local training costs for clients.***

### 2.1 Federated Learning

FL is a paradigm for iteratively training a machine learning (ML) model using large amounts of training data distributed across hundreds to potentially millions of client devices [34]. An FL framework comprises two main entities: a centralized server or *aggregator* and distributed devices or *clients*. The training process is spread over multiple rounds, typically on the order of hundreds, to improve model accuracy by leveraging large numbers of clients, as discussed below [5].

In each round, the aggregator decides on the *number of clients* and the *specific clients* that will participate in that round. The aggregator then sends the latest model parameters and configuration, e.g., hyper-parameter settings, to the selected clients, called *participants*. Each participant trains the model on its local data using the specified hyper-parameters. At the end of local training, the participant computes the model update, which specifies the delta from the model parameter it received at the start of the round. The client then sends the model update to the aggregator.

Upon receiving model updates from all participating clients, the aggregator updates the global model based on an aggregation strategy, such as FedAvg [5]. Importantly, model updates from all participating clients may not arrive simultaneously due to differences in participant local training times and network delays in their path to the aggregator. Depending on the local data characteristics and hardware heterogeneity, the local training time can vary widely across participant clients. Thus, the aggregator typically waits for a specified amount of time or number of clients before aggregating the model updates to the global model.

FL generally operates as a continuous learning environment, where training data flows in constantly. In such a dynamic setting, a pivotal question is when and whether model retraining is necessary and how data should be selected. In

FL, determining which data to use is part of the client selection policy. The decision to initiate a new training round hinges on a stopping criteria. Prior work uses various criteria to determine when further federated learning rounds are no longer required. Each stopping criterion can have distinct implications for both cost and accuracy. In this paper, we focus on two widely-used stopping criteria: i) *model convergence*, where training stops when minimal improvements in model performance are observed; and ii) a *performance threshold*, where training halts when the model's performance on the validation dataset reaches a predefined threshold.

### 2.2 Resource-efficient FL

FL's resource efficiency is dictated by the operation of both the aggregator and the clients. While significant efforts have been directed at optimizing the infrastructure on which aggregators operate (e.g., [18]), FL essentially delegates a substantial portion of the computational load required for model training to the clients. These *clients* are tasked with retraining the model received from the FL server using their local data, a process that consumes both computational and communication resources, translating into a measurable *cost*.

Notably, FL has made progress in optimizing model performance [31] and system efficiency [34] to address challenges arising from data and system heterogeneity. However, the potential for mitigating these challenges by selectively choosing participants before execution has only recently received significant attention. Traditionally, FL training and testing have heavily relied on random participant selection [5, 21].

Recent research has made significant strides in improving resource utilization compared to random selection. For instance, Oort [28] employs a guided participant selection algorithm that prioritizes learners with higher statistical utility while maximizing system efficiency. Statistical utility is gauged using training loss as a proxy, while system efficiency is quantified as a function of completion time. Oort strategically favors faster learners to reduce round duration and employs a pacer algorithm that employs a longer round duration to include unexplored or slower learners to improve total statistical efficiency. A recent extension of this work, REFL [1], focuses on minimizing resource waste by reducing the selection of straggler clients that perform computations but are ultimately not utilized by the central model.

Additionally, other approaches, such as PyramidFL [29], aim to fine-tune selection criteria by considering not only data and system heterogeneity between all clients but also heterogeneity within the selected clients. As we discuss, CEFL's approach focuses on client selection, similar to Oort but with an explicit focus on cost, while both REFL and PyramidFL are optimizations orthogonal to our work that could improve CEFL's resource efficiency.

CEFL recognizes that resources and costs can differ across clients. To shed light on these differences, we assess the impact of data heterogeneity, especially when costs vary across clients. Beyond cost considerations, CEFL leverages recent insights into the significance of critical learning periods [2, 44]. In summary, CEFL distinguishes itself from other resource-efficient approaches in two fundamental ways: first, it acknowledges that client costs need not be equal, as shown using carbon emissions as an illustrative example throughout the paper. Second, CEFL treats the number of clients selected, not as a fixed parameter, but as an important variable. Surprisingly, this approach can decrease costs by notably increasing the number of clients in the initial rounds.

| **Metric** | **Description** | **Unit** (total or per-round) |
|---|---|---|
| **Time** | Time to convergence or accuracy. | *seconds* |
| **Energy** | Total amount of energy use. | $kWh$ |
| **Price** | The monetary cost of training, due to energy use, battery degradation, etc. | $ |
| **Carbon** | The amount of carbon emissions. | $g \cdot CO_2eq$ |
| **State-of-Charge** | The amount of charge used or the amount of charge left. | % |

**Table 1.** ***Potential cost metrics for the clients in FL.***

### 2.3 The Notion of Cost in CEFL

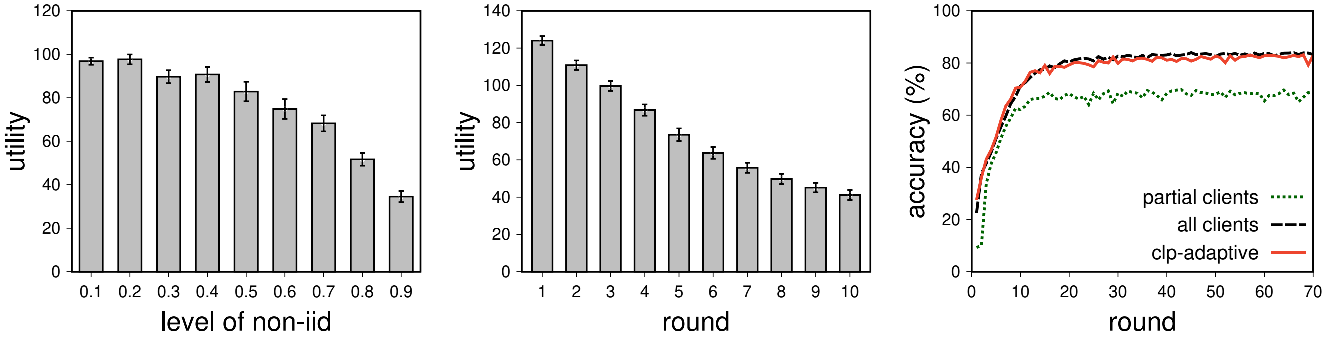
The traditional cost metric used in FL is *time*, including time-to-convergence or time-to-accuracy. While time is an important metric, it does not encompass the other potential costs that clients training locally may incur, e.g., energy and carbon. As illustrated in Figure 1, prior research focuses on enhancing system efficiency by addressing stragglers to decrease the energy consumed by the stragglers that do not ultimately contribute to the trained model. This research is orthogonal to our goal of reducing the cost of local training.

In practice, there are many possible cost metrics that may differ widely across different users. For example, the amount of energy consumed by a client when training in a round may differ based on local data and hardware characteristics, the dollar cost of energy may differ based on the location, and the carbon emissions of training may differ due to differences in the local energy generation mix.

Table 1 lists potential cost metrics that FL clients might consider important during training. A battery's state-of-charge may be defined in multiple ways: for example, clients may assign a higher value to the last 20% of a battery's state-of-charge than the first 20%. In this work, we use carbon as our primary illustrative cost metric. We analyze the spatiotemporal characteristics of carbon as a cost metric for CEFL in Section§5.1. Note that our list of cost metrics is not complete, and CEFL can incorporate any arbitrary cost metric (static or dynamic) on a per-client basis.

### 2.4 Cost-Efficient Client Selection

Prior work has proposed various client selection strategies, beyond random selection, that target an exogenous objective, such as maximizing fairness, reducing time-to-accuracy, and improving robustness. Prior work of Németh et al. [35] presents a survey of client selection strategies in FL.



(a) utility vs. level of non-iid (b) utility vs. no. of rounds (c) impact of critical learning period

**Figure 2.** ***The statistical efficiency of clients ( utility) decreases with the increase in the data heterogeneity across clients, quantified using the level of non-iid, (a) and the number of training rounds (b). The impact of critical-learning-aware client selection approach on the accuracy of the trained model.***

In our work, we decompose the cost-efficient client selection problem into two steps: first, determining the number of clients in each round [44], and second, choosing the specific set of clients for a given round [28]. As mentioned before, while there is prior work on determining which and how many clients to select for each round, they have been employed individually and in a cost-agnostic manner. In Section 3, we describe how we extend and combine this prior work to devise cost-efficient client selection policies for CEFL. Here, we describe the prior work in detail and highlight its core characteristics that inform CEFL.

We first look at how the statistical utility of the clients varies across different settings and stages of the training in FL. In each round, the statistical utility [28] is calculated by computing the gradient of the loss, which is obtained by taking the derivative of the training loss with respect to the current model weights. The training loss quantifies the discrepancy between model predictions and ground truth. Therefore, the statistical utility $U(i)$ for client $i$ is defined as:

$$U(i) = |B_i| \times \sqrt{\frac{1}{|B_i|} \sum_{m \in B_i} \mathrm{Loss}(m)^2}$$

where $|B_i|$ denotes the size of client $i$'s training samples, and $\mathrm{Loss}(m)$ represents the training loss for sample $m$.

Figure 2(a) shows the magnitude and variance in the client's statistical utility based on the data distribution. In general, if all clients have the same distribution of classes, the data distribution is identical and independently distributed, or iid. If clients have different proportions of classes, it is non-iid. In Section 5, we describe an approach for precisely controlling the data distribution such that 0 indicates iid data, i.e., all clients have the same classes of data in the same proportion, 1 indicates entirely non-iid data, i.e., each client has only a single class of data and no others, and the numbers in between define a range between these two extremes.

Figure 2(a) shows that the average utility of clients decreases as the extent of non-iid increases because the clients have more highly specific data, and thus their model updates are less helpful in improving the accuracy evaluated against the complete dataset. Figure 2(b) then shows the magnitude and variance in clients' statistical utility over different training rounds. As the model is trained, the data on each client becomes less useful, and its model updates contribute less towards improving the global model's accuracy. Note that the utility across clients for the same level of non-iid for a given round does not vary significantly. This observation suggests that a client selection policy may prefer to select high-cost clients during the earlier training rounds when their utility is high and vice versa.

The critical learning period (CLP) represents the initial training rounds that bootstrap the model weights, which significantly influences time-to-accuracy [44]. Aggressively selecting clients during this period can notably improve time-to-accuracy, as shown in prior work. Figure 2(c) shows the accuracy achieved for various client selection policies. As we observe, the final accuracy a model is able to achieve significantly degrades if only a partial set of clients is selected in each round. However, a critical learning period-based policy can compensate for the loss in accuracy by choosing a higher number of clients during the critical learning phase while significantly reducing them afterward. In this case, the final accuracy is the same as selecting all the clients in all rounds, but the average number of clients across all rounds is the same as the policy that selects only partial clients for all rounds. This integration presents an opportunity to further reduce the overall cost of federated learning (FL) by dynamically adjusting the number of clients during training, increasing the number of selected clients during the CLP, and dropping a certain number of clients after the CLP ends.

## 3 CEFL Design

Given a set of clients that each have a variable cost, CEFL's goal is to minimize the cost of FL training to reach a target accuracy by determining which and how many clients

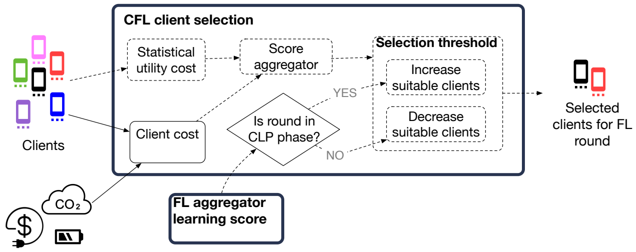


**Figure 3.** ***CEFL framework.*** **Dotted lines indicate that communication or computations happen frequently, e.g., every round; solid lines indicate that communication or computation occurs once or upon request.**

to select during each round of training, as well as when to select them. We assume the per-client cost is static and known *a priori*, and is a function of the local computation. We illustrate cost using carbon emissions, as energy's average carbon-intensity (in g·$CO_2$/kWh) varies widely across clients in different geographic locations. In this case, selecting a client for training increases its energy consumption which is multiplied by its energy's average carbon-intensity to yield its carbon emissions. In this example, CEFL's goal is to minimize FL's carbon emissions to reach a target accuracy.

### 3.1 CEFL Overview

Figure 3 illustrates CEFL's client selection process. The process begins by gathering client cost data, which may come from clients directly or via third-party sources, e.g., [33]. In many cases, as with carbon emissions, the cost may remain static for long periods. In parallel, the system estimates the expected statistical data utility of each client. This information feeds into a combined score aggregator, which combines both cost and statistical utility, with a focus on the utility-per-cost. At the same time, the FL aggregator shares its learning score information, which plays a crucial role in determining whether the current round falls within the CLP. In CLP rounds, there are multiple potential policies for increasing client selection, with priority given to clients offering higher utility-per-cost, as provided by the score aggregator.

In FL, the process of client selection is akin to assembling an elite team for a critical mission, where the objective is to achieve an optimal balance between cost efficiency, performance, and the learning period. Efficiency involves minimizing environmental impact by selecting energy-conscious clients to reduce carbon emissions (other costs can also be considered, refer to Table 1). Performance focuses on maximizing the accuracy of the global model, such that we carefully choose clients that can contribute to this objective. Further, we recognize the significance of the learning period; FL exhibits critical learning periods during which even minor gradient errors can have an irreversible impact on the final test accuracy. In this section, we discuss strategies for optimizing client selection based on these criteria.

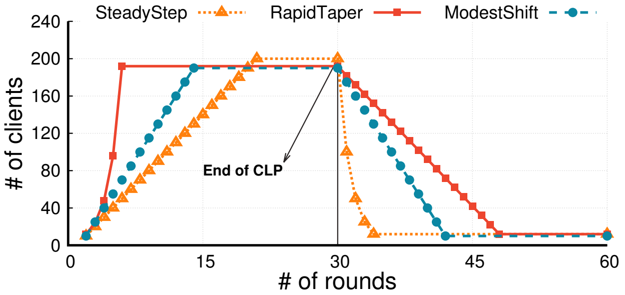


**Figure 4.** ***Variants of client scaling strategies in CLP.***

**Cost-based selection.** We start by defining and understanding guided client selection where clients with the lowest cost are chosen. This approach reduces the overall cost of the FL training process compared to utility-based and random-based selection policies. As discussed in (§2.2), the selection decision is made based on a specific cost metric, which, in our case, is the carbon-intensity ($g \cdot CO_2eq/kWh$ ) associated with each client's local training (assuming that clients consume the same energy per training round).

We utilize real-world carbon intensity traces from 123 distinct geographical locations between 2020 and 2022, calculating the average carbon-intensity per location [33]. We randomly assign this cost, in terms of carbon-intensity, to a client for its participation in the FL training. Consequently, at the start of each round, CEFL selects clients with the lowest carbon-intensity values. To prevent repeated selections of the same set of low-cost clients throughout the training process, we restrict CEFL from choosing a specific client more than $\epsilon$ times, which can be configured by the user.

**Utility/Cost based selection.** Although the cost-based approach effectively reduces the overall carbon footprint of the training process, selecting clients based solely on cost can introduce bias into the model [47], impacting the overall model accuracy. A client selection approach relying solely on cost is expected to perform worse than both random and utility-based approaches. To address this concern, we introduce a new metric that combines utility and cost considerations when selecting clients for the next training round. We define this metric as the ratio of utility to cost for a client $i$, denoted as $U_i/C_i$, where:

$$U_i/C_i = \frac{|B_i|}{C_i'} \times \sqrt{\frac{1}{|B_i|} \sum_{m \in B_i} \text{Loss}(m)^2}$$

Here, $C_i'$ represents the average carbon intensity associated with location $x$, randomly assigned to client $i$, then, $B_i$ are all the training samples of client $i$ and $|B_i|$ denotes the total number of training samples from client $i$.

At the start of the training process, CEFL selects $K$ clients based on the lowest cost values, since determining client utility before their participation is unfeasible in realistic FL settings where there is a large number of clients. In subsequent rounds, CEFL utilizes the $U(i)/C(i)$ metric to select clients, employing the exploration-exploitation technique.

**Algorithm 1:** Algorithm for detecting critical learning period.

```
1 input: accuracy curve $A(r)$ where $r$ is the round number
2 input: $T$ threshold for the derivative magnitude;
3 $A'(r) \leftarrow \frac{dA}{dr}$;
4 $mu \leftarrow |A'(r)|$;
5 $mu_r \leftarrow \frac{1}{w}\sum_{i=r-w+1}^{r}|A'(i)|$;
6 if $mu_r < T$ then
7     return True;
8 return False
```

Specifically, $e \times K$ clients are explored, where $e \in [0, 1)$, and $(1 - e) \times K$ clients with the highest $U(i)/C(i)$ - indicating both high utility and low cost - are selected. As in the cost-based selection, CEFL offers a configurable parameter, ($\epsilon$), to limit the number of times a client can be selected.

### 3.2 CEFL's Client Selection Policy

CEFL integrates the client selection policies with the critical learning period (CLP) theory to enhance time-to-accuracy and cost-efficiency. The CLP represents the initial training rounds that bootstrap the model weights, significantly influencing time-to-accuracy. Aggressively selecting clients during this period can notably improve time-to-accuracy. This integration presents an opportunity to further reduce the overall cost of FL by dynamically adjusting the number of clients during training, increasing the number of selected clients during the CLP, and dropping a certain number of clients after the CLP ends.

In our system design, we address the following key questions regarding the CEFL's client selection policy:

1. *Which* clients to select during the CLP and which to drop after the CLP ends?
2. *How many* clients to select during the CLP and how many to drop after the CLP ends?
3. *When* to drop the clients, i.e., how to determine when the end of CLP is reached?

**Which clients does CEFL's client selection policy choose?** The client selection in CEFL aims to maximize both utility towards model performance and cost-efficiency during training to achieve the desired accuracy. Clients are chosen based on the highest ratio of utility/cost, to reduce time-to-accuracy and overall training costs. The selection policy is dynamic, with the number of clients chosen varying each round to adapt to the training process.

**Dynamic client selection.** At the start of the training, CEFL's policy initiates the selection of clients in the first round to assess client utility values. The process starts by selecting $M \times N$ clients, where $M$ is an integer value within the range $[0, 1]$, and $N$ denotes the total number of clients to be selected in the round. Subsequent rounds follow the utility/cost strategy, dynamically adjusting the number of clients chosen for the next round, giving priority to clients with higher utility/cost values, while following the exploration and exploitation strategy previously described, more details of the evaluation methodology can be found in (§5.1).

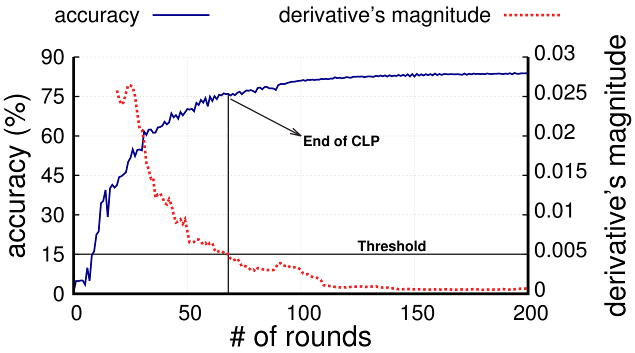


**Figure 5.** ***Finding the critical learning period (CLP) dynamically based on the derivative of accuracy.***

**Number of clients during and after CLP.** We define three policies to determine the rate at which clients are added and during the critical learning period (CLP) and dropped after:

1. **SteadyStep Policy:** Clients are added steadily during the CLP (increased by $\alpha \times N$) and dropped steeply (by a factor of $\beta$) after the CLP.
2. **RapidTaper Policy:** Clients are added rapidly during the CLP (increased by a factor of $\alpha$) and dropped slowly (reduced by $\beta \times N$) after the CLP.
3. **ModestShift Policy:** Clients are added moderately during the CLP (increased by $\alpha \times N$) and dropped moderately (reduced by $\beta \times N$) after the CLP.

We used $\{\alpha, \beta\}$ values equal to $\{0.01, 2\}$, $\{2, 0.01\}$ and $\{0.015, 0.015\}$ for SteadyStep, RapidTaper, and ModestShift policies, respectively. Figure 4 illustrates how clients are selected by these strategies; note that the *RapidTaper* policy closely resembles the original CLP proposed in [44].

**When to drop the clients?** As previously discussed, CEFL employs a selection policy that drops clients when the CLP phase concludes. This is because the global model's learning capability diminishes after the conclusion of CLP, and not all clients are required in the training process at that point. In this context, we attempt to dynamically determine when the CLP terminates to construct a closed-loop system.

In comparison to prior work, which emphasizes analyzing gradients of loss acquired during local client training and subsequently aggregating weighted gradient losses from all clients, our method directly references the accuracy of the global model (input of Algorithm 1, line 1). This accuracy is the outcome of predictions made using the global model parameters resulting from the aggregation of local parameters from all clients, utilizing the FedAvg FL strategy [5]. We then compute the derivative of the accuracy curve (line 2) and calculate the magnitude of this derivative (line 3), subsequently smoothing it using a moving average with a window size, $w$. This process helps in quantifying trends or alterations in accuracy during the training process.

Typically, accuracy exhibits a swift increase during the initial learning phase. Consequently, in this phase, within the CLP, the derivative values should be elevated, reflecting the steep slope of the accuracy curve. This slope gradually reaches its maximum and then declines, all within the CLP, coinciding with the point at which the accuracy curve begins to flatten out after some number of rounds. As the magnitude of the derivative value falls below a predefined threshold established by CEFL, the system marks that particular point as the conclusion of the CLP. See Figure 5.

Thus, we address all three design questions regarding the client selection policy within CEFL in this subsection. Consequently, the client selection policy in CEFL is based on a consideration of cost and utility, incorporating a dynamic client selection process during the CLP. Thus, optimizing the trade-off between overall cost and accuracy, as well as the time required to achieve the desired accuracy.

## 4 CEFL Implementation

Our implementation leverages Flower framework [4], which provides a basis for federated learning (FL) orchestration, facilitating communication between the global server and geographically distributed clients. Flower uses PyTorch as the underlying ML library. To emulate FL within Flower, we utilize its Virtual Client Engine (VCE), allowing effective configuration and management of a cluster of servers for rapid execution of federated learning workloads. The clients managed by VCE possess the following key characteristics:
**Resource awareness.** Clients are allocated specific compute and memory resources. This allocation is configurable at the onset of the experiment, granting control over the degree of parallelism in the Flower FL emulation. Higher client concurrency is achieved by limiting resources per client.
**Self-management.** Virtual Client Engine internally handles client instantiating, obviating the need for manual launching.
**Ephemeral nature.** Clients are instantiated only when necessitated by the federated learning process, such as during model fitting. Post-operation, the client instances are promptly de-materialized, relinquishing allocated resources and allowing other clients to partake.

We extended the Flower framework to align with our specific federated learning requirements. These extensions encompass significant enhancements to the global server program, an extended federated learning strategy for identifying critical learning periods, custom configurations for controlling client selection and dropping in each federated learning round, improvements to the client manager for intelligent client selection, and the definition and integration of a client selection policy based on various metrics, including utility, cost, and cost/utility.

**Table 2.** ***Statistics of datasets used in the evaluation.***

| Dataset | Images | Classes | Image Size | Balanced | Simulated Clients |
|---|---|---|---|---|---|
| CIFAR10 | 60,000 | 10 | 32×32 RGB | yes | 500 |
| SVHN | 99,298 | 10 | 32×32 RGB | yes | 200 |
| EMNIST | 814,255 | 62 | 28×28 grey | no | 1000 |

In addition, we have implemented an automated monitoring system for the training process, enabling real-time tracking of client removals during training rounds. This system captures extensive metadata related to the global model's learning progress, encompassing metrics such as loss, accuracy, training time per round, and the environmental footprint of the learning process. This data is recorded to facilitate in-depth analysis and evaluation. We implemented CEFL with ≈2000 lines code on top of Flower. We will publicly release our code along with the publication of this paper.

## 5 CEFL Evaluation

In evaluating CEFL, we answer the following questions.

1. How do cost-aware client selection policies compare to the existing random and cost-agnostic policies? (§5.2)
2. Does incorporating the critical learning period into cost-aware client selection offer a better trade-off? (§5.3)
3. How does CEFL generalize across different datasets? (§5.4)

### 5.1 Evaluation Methodology

Below, we present our evaluation methodology, including (1) the hardware used for experiments, (2) training datasets, models and their parameters, (3) characteristics of carbon as a cost metric, and (4) the baseline policies.
**(1) - Experimental setup.** We evaluate CEFL on a small-scale cluster of 40 `NVIDIA GTX 1080Ti`, 20 `NVIDIA RTX 2080Ti`, and 6 `NVIDIA RTX 8000` GPUs to emulate a fairly large-scale FL scenario. We emulate up to 1,000 participants in the FL training process, representing large-scale edge deployments using the "simulation" module within Flower [4].
**(2) - Datasets and models.** We conduct experiments with three widely-used datasets, each presenting different characteristics in terms of scale and complexity: CIFAR10 [27], SVHN [36], and EMNIST [10]. Table 2 summarizes the key characteristics of the datasets used. Both the CIFAR10 and SVHN datasets are balanced in terms of classes, i.e., all classes have the same number of images. EMNIST is an unbalanced dataset consisting of 28×28 greyscale images.

We distributed the EMNIST dataset across 1000 clients since it is a larger dataset with more than 800k samples in both the training and testing sets. For CIFAR10 and SVHN, which are smaller and have fewer classes, we distributed the dataset among 500 and 200 clients, respectively. However, such distribution only provides an independent and identical distribution (iid) in terms of data, i.e., all the clients have the same distribution of classes. To simulate real-world FL settings, we also change the distribution of the data across clients to generate various non-iid levels.

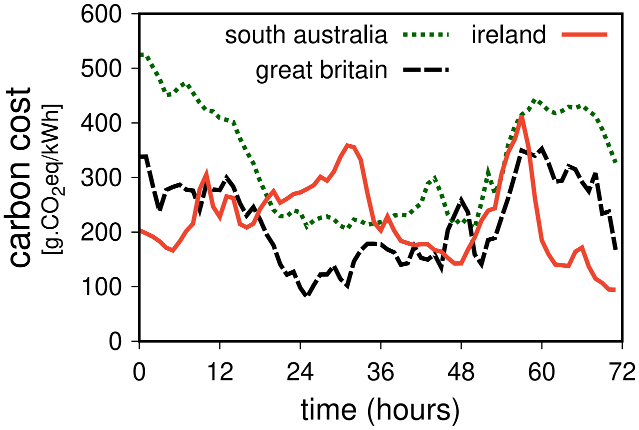


**Figure 6.** ***Carbon cost for three example geographically distributed federated learning clients.***

**Achieving non-iid data distributions.** We leverage the Fang distribution [13] to create the non-iid characteristics in the dataset. We control the level of "non-iid*ness*" using a knob value that ranges from 0 (fully iid) to 1 (fully non-iid). The non-iid knob value influences the distribution such that at a value of 1, a single client would represent a single class within the dataset, while having no representation of other classes. The value 0 represents fully iid data, such that all clients receive the data in equal proportions. This approach enables us to precisely vary data heterogeneity across clients.
**DNN parameters.** We use the ResNet-18 architecture for training, a widely-used deep neural network model for various computer vision tasks. In each training round, we select 10% of the clients as the default unless stated otherwise. The initial learning rate was set at 0.01. The batch size was 20 for EMNIST and 32 for CIFAR10 and SVHN. In the exploration-exploitation phase of training, inspired by Oort, we use an exploration factor of 0.1, i.e., 10% previously unexplored clients are included in each round. We used 2 epochs for CIFAR10 and 1 for SVHN and EMNIST datasets. We use FedAvg as the aggregation strategy at the global server model for the aggregation of clients' parameters for all datasets.
**(3) - Carbon as a cost metric.** We use electricity's carbon-intensity as our example cost metric. Carbon-intensity is the mass of carbon emitted per unit of consumed electricity, generally measured in grams of carbon dioxide equivalent per kilowatt-hours ($g \cdot CO_2eq/kWh$ ). The carbon-intensity of grid-supplied electricity depends on the mix of generators used to satisfy demand. The grid's energy demand varies primarily based on weather, which dictates the energy needed for indoor heating and cooling, and human behavioral patterns, e.g., time of the day, day of the week, holidays, etc. The mix of generators used to satisfy a variable demand changes over time, resulting in carbon-intensity changes.

In general, the carbon cost in the 123 regions we consider for clients varies widely from the lowest value of $14.96g \cdot CO_2eq/kWh$ for Sweden and the highest value of $947g \cdot CO_2eq/kWh$ for Cyprus, with a global average of $369.74g \cdot CO_2eq/kWh$ . To illustrate, Figure 6 shows the carbon cost for three example regions over a three-day period.
**(4) - Baseline policies.** Below, we present the baseline approaches that we compare CEFL against.
**Random selection.** Random client selection involves the selection of clients from the available client pool in a random manner to participate in a specific round of training. This randomness aims to prevent any particular client or group of clients from dominating the training process, promoting fairness and minimizing bias. However, random client selection can extend the training process due to its indiscriminate selection of clients, including those with fewer samples or noisy data. As a result, the model learns slower and takes longer to achieve a specific accuracy, potentially requiring a larger number of training rounds to cover the entire dataset.

For our baseline, we adopt a random selection strategy by choosing $K$ clients out of the total number of $N$ clients for each round of training based on a uniform data distribution, where $K$ is defined as $10\% \times N$. This random selection serves as the baseline for comparison with cost-aware policies, and we follow the same rule for selecting $K = 10\% \times N$ number of clients per round in the rest of the policies discussed below.
**Utility based selection.** Utility-based selection is a guided client selection strategy where the clients are selected in each round of training based on their high statistical utility towards the model performance instead of selecting clients randomly [28]. Statistical utility refers to the probabilistic measure of a client's contribution towards improving the model performance for various tasks while respecting privacy. This approach improves the time to reach a specific accuracy since the model is fed with client data, contributing the most towards improving the performance. In terms of time to accuracy, it outperforms random selection.

The statistical utility, $U_i$, can only be determined after its participation in training. However, when dealing with a large number of clients, trying each client individually becomes impractical. To address this challenge, we use an *exploration-exploitation technique*, as in Oort [28]. This technique leverages clients that have previously been selected and whose utilities have been computed, while also exploring new, unexplored clients to identify those with high utility.

At the start of each round, the client manager receives the client utilities from the previous training round and updates the statistical utility and system performance for each client. For clients that have been explored previously, the system calculates their client utility, enabling the *exploitation* of high-utility participants. The *exploration* of clients is performed randomly, and an exploration factor, $e$, governs the proportion of exploration. For instance, if there are $N$ total clients and $K$ clients are to be selected in each round, the client manager chooses $e \times K$ explored clients and $(1-e) \times K$ clients for exploitation. To prevent repetitive selection of the

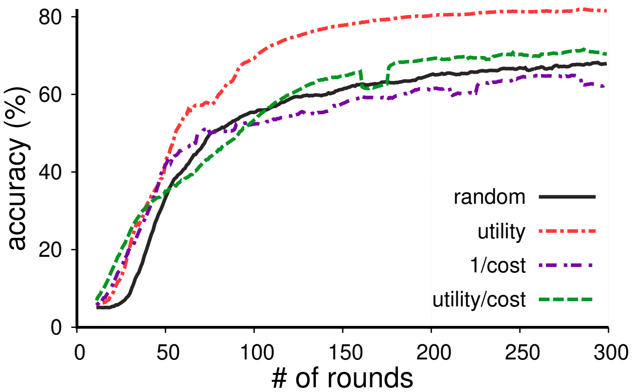


**Figure 7.** ***Accuracy curves for baselines and our policies.***

same high utility clients throughout the training process, we restrict CEFL from choosing a specific client to no more than 10% of the total number of rounds.

### 5.2 Cost-aware Client Selection

Below, we evaluate the cost-aware client selection policies i.e., cost- and utility/cost-based selection policies against the random and utility based selection baselines. Here, we focus on: 1) the impact of cost-aware policies on the time and cost to reach a target accuracy, where we choose the convergence accuracy of random selection as baseline and 2) exploring these policies across different levels of data heterogeneity.

**Convergence Accuracy.** Figure 7 illustrates the model performance for various client selection policies at **non-iid - 0.9**. The x-axis represents the time in terms of the number of rounds required to achieve convergence accuracy, while the y-axis denotes the model accuracy for different selection policies. The baseline strategies—utility-based client selection and random selection—are depicted by the dotted red and solid black lines, respectively. The cost-aware selection policies i.e., cost-based and utility/cost-based are represented by the dotted purple and dashed green lines, respectively. This result demonstrates that the cost-based selection approach performs the worst, as it solely selects clients based on the lowest carbon-intensity values. This may lead to the repeated selection of the same low carbon cost clients, thereby penalizing model performance. Random selection outperforms the cost-based approach by introducing randomness in client selection each round, mitigating the bias introduced by selecting the same clients repeatedly. The utility/cost-based selection performs better than both random and cost-based approaches, as it leverages both the utility of clients in each round and their low cost for selection. Finally, the utility-based approach excels in terms of accuracy. Figure 8, non-iid - 0.9 illustrates that the utility/cost-based approach gives ***7.5%*** better convergence accuracy than the simple 1/cost-based and ***3.1%*** better than random approach.

**Cost-to-accuracy.** Figure 9, non-iid - 0.9, shows the carbon cost associated with the training of the federated learning model for a specific number of rounds needed to attain the target accuracy, specifically the convergence accuracy of the random selection baseline. The figure illustrates that the utility/cost-based approach surpasses all other policies, reducing carbon consumption by ***92.33%*** compared to random selection and by ***78.86%*** compared to utility-based selection.

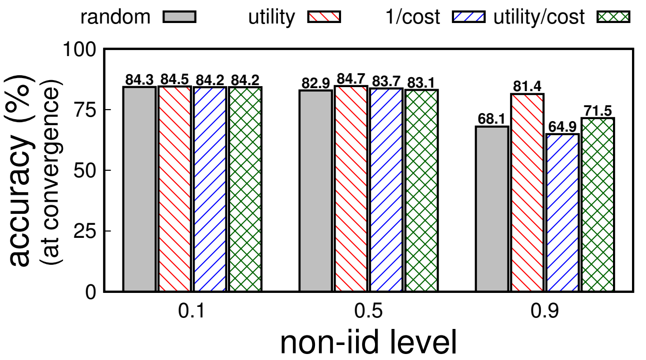


**Figure 8.** ***Effect of different non-iid levels on the performance of the global model under different client selection policies for the EMNIST dataset.***

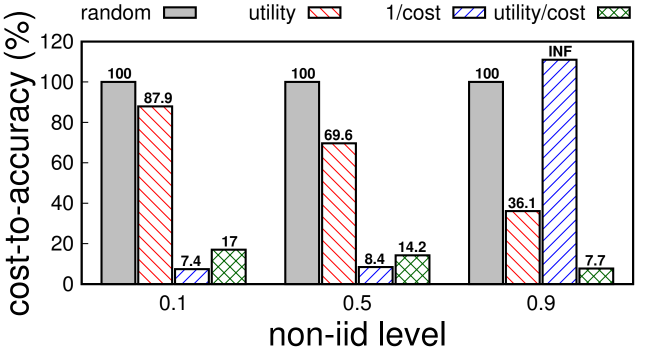


**Figure 9.** ***Carbon cost for number of rounds to reach the convergence accuracy of random selection.***

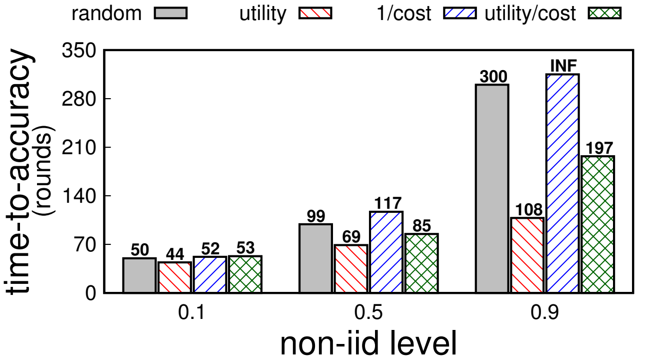


**Figure 10.** ***Time-to-accuracy (quantified using no. of rounds) for various client selection approaches.***

**Time-to-accuracy.** Figure 10, non-iid - 0.9, compares the time-to-accuracy (convergence accuracy of random selection) of the cost-aware selection policies against the random and utility-based selection baselines. The graph presents the rounds required (y-axis) for the selection policies (x-axis) to reach the convergence accuracy of the random baseline.

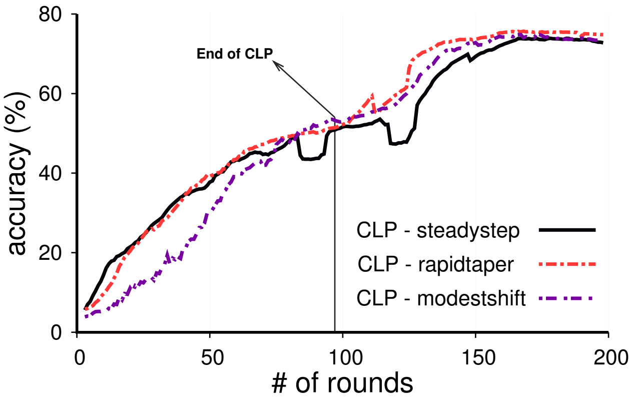


**Figure 11.** *Accuracy for various CLP-based strategies.*

Our analysis demonstrates that the utility-based approach performs notably faster than any other policy, requiring approximately *64%* fewer rounds than the random baseline. The utility/cost-based approach achieves the next best accuracy, requiring around *34%* fewer rounds. In contrast, the cost-based selection never converges to the same accuracy as the random baseline, making it the least efficient.

**Key Point.** *Our analysis shows that the utility/cost client selection policy outperforms all other selection policies, including random, utility, and cost-based selection policies, in terms of the carbon cost to reach a target accuracy.*

**Effect of Data Heterogeneity.** Figure 8 shows the effect of varying non-iid levels in the data distribution on our selection policies. Lower non-iid values imply a more similar data distribution across clients, indicating that the utility of all clients is nearly identical. At low non-iid values (0.1), the final model accuracy across all selection policies is similar. In this scenario, utility-based selection has minimal impact since all clients contribute equally to the model's performance.

The model's convergence accuracy remains comparable as we increase the non-iid level to 0.5. However, Figure 10 shows a significant change in the number of rounds it takes to reach the convergence accuracy of the random selection baseline. The utility-based approach requires the fewest rounds (approximately *30%* fewer rounds than random), followed by utility/cost (approximately *15%* fewer rounds than random), while the cost-based approach performs the worst in terms of time (number of rounds) (*17%* more rounds than random) to accuracy. Additionally, Figure 9 illustrates the carbon cost for achieving the baseline (random) accuracy. The cost-based approach demonstrates a direct relationship between cost and non-iid level, indicating that higher non-iid values increase the cost to reach a target accuracy, given the extended convergence time. Despite the utility/cost approach requiring approximately *40%* more cost to reach a specific accuracy than the cost-based approach at non-iid 0.5, it still yields significant savings of approximately *80%* and *85%* compared to utility and random-based selection, respectively. Moreover,

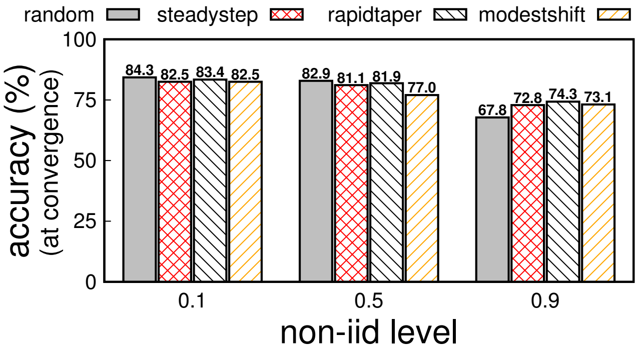


**Figure 12.** *Accuracy of client scaling strategies for CLP.*

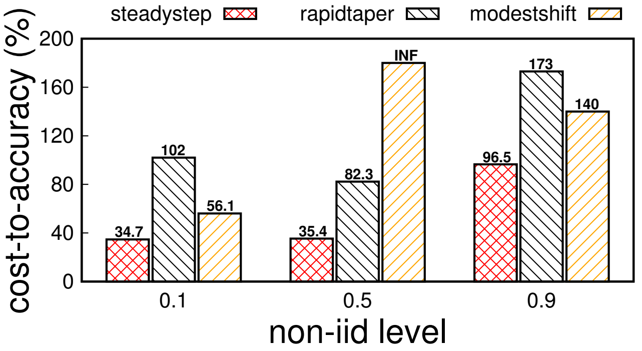


**Figure 13.** *Cost to reach the same accuracy as random selection policy for different client selection strategies within critical learning period.*

it follows the right trends by taking fewer rounds to reach accuracy with increasing levels of non-iid.

At high non-iid values, these trends become even more pronounced, as already discussed. The cost-based approach converges to a lower accuracy than random selection, requiring an infinite amount of time to achieve the same accuracy. The utility/cost-based selection approach outperforms the rest in terms of cost to reach a specific accuracy, as shown in Figure 9 at high non-iid values, providing approximately *92%* cost reduction while requiring approximately *34%* fewer rounds than the random baseline.

### 5.3 Cost Reductions using Critical Learning Period

In this section, we perform a cost analysis on different client selection policies with respect to the critical learning period (CLP) as outlined in (§3.2), where we discussed the significant impact of the critical learning period (CLP) on the FL model's training. Thus, we evaluate our integrated CLP and utility/client selection strategy, which further reduces the cost to achieve a specific accuracy. We first select the steadystep policy within CLP to analyze cost and accuracy to compare with the other cost-aware and baseline policies at non-iid - 0.9. Recall that the steadystep policy works by adopting a steady increase in selected clients within CLP, followed by a halving of clients post-CLP. In Figure 11, the black solid line refers to the accuracy of steadystep policy

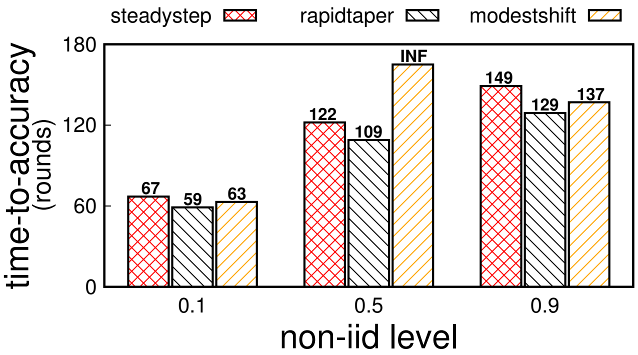


**Figure 14.** ***Time (number of rounds) to reach the same accuracy as random selection policy for different client selection strategies within critical learning period.***

within CLP against the number of rounds, while the arrow marks the end of the CLP, which occurs at round number 97. In CEFL's client selection policy based on CLP, utility and cost reaches convergence to the random baseline accuracy in 149 rounds reducing the convergence time (number of rounds) further by *24%* (shown in Figure 14) while maintaining the cost (approximately 2% less than utility/cost) compared to standard non-CLP utility/cost-based selection (shown in Figure 13). Overall, CEFL's client selection policy takes approximately *50%* fewer rounds and approximately *93%* less in cost compared to the random baseline, demonstrating the potential of CLP in substantially minimizing the cost and time of achieving a specific accuracy in FL training.

**Effect of Scaling Strategies in CLP.** Figure 12 shows that the *RapidTaper* accuracy performs slighly better than *SteadyStep* and *ModestShift*. This is primarily due to its ability to retain a significant client count for extended training durations. Such a high number of clients causes it to incur the highest carbon cost during training due to its swift increase in the number of clients within CLP, maintaining a large client base for an extended duration. Conversely, the *SteadyStep* strategy adopts a more steady approach by gradually increasing the number of clients within CLP and rapidly reducing them post-CLP. This rapid reduction in the client count after CLP significantly reduces associated costs by minimizing the period during which a large client base is selected for training. The accuracy performance of the *SteadyStep* policy remains comparable to that of *RapidTaper*.

The *ModestShift* strategy adopts a more conservative approach in both client ramp-up during CLP and subsequent reduction post-CLP. Consequently, it incurs intermediate costs, positioning it between the *RapidTaper* and *SteadyStep* policies. In Figure 14, we observe that at higher non-iid value (0.9) *SteadyStep* takes more rounds to reach the convergence accuracy of random baseline compared to others, *RapidTaper* and *ModestShift*. However, it still takes approximately ***50%*** fewer rounds than random selection and approximately ***24%*** fewer rounds than non-CLP standard utility/cost policy.

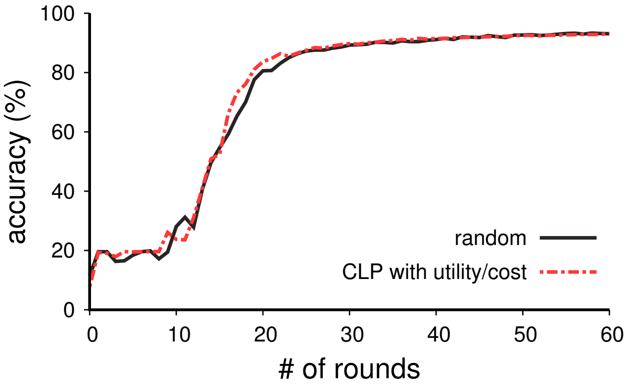


**Figure 15.** ***SVHN: Random vs CLP utility/cost policies. CLP saves costs while achieving the same accuracy.***

In terms of cost to reach this accuracy, *SteadyStep* outperforms not only *RapidTaper* and *ModestShift* but all the other non-CLP client selection policies as well (approximately *93%* lower cost compared to the random selection). Similarly, at lower non-iid levels, CEFL's utility/cost-based selection with steadyshift results in *94%* cost reduction compared to random baseline, which is the most cost reduction among any other non-CLP policy including the cost-based selection.

### 5.4 Evaluating CEFL Generalization

We next evaluate CEFL's generalization by applying it to multiple datasets with different characteristics. Specifically, we evaluate CEFL's performance across CIFAR10 and SVHN at a moderate non-iid (data heterogeneity) level of 0.5.

**SVHN.** Figure 15 illustrates the convergence accuracy for the random client selection and CEFL's client selection policies with the SteadyStep selection strategy within the critical learning period. We observe similar convergence accuracy for random and CEFL. Table 3 shows CEFL achieves a cost reduction of 76% for the same accuracy as a random baseline.

**CIFAR10.** We also evaluate the performance of CEFL on the CIFAR-10 dataset. We compare CEFL's client selection policy, specifically the utility/cost based selection and SteadyStep scaling strategy within CLP, with a random client selection policy. Figure 16 shows the convergence accuracy of both policies where the two policies exhibit similar performance in terms of convergence accuracy. Table 3 shows that CEFL outperforms random selection by reducing the number of rounds required to reach the target accuracy by approximately 26% and the associated cost by approximately 57%.

**Key Point:** *Our results on two different datasets suggest that CEFL's cost and accuracy tradeoffs are generalizable.*

## 6 Related Work

**Non-IIDness.** FL operates within the domain of privacy-preserving distributed algorithms[16, 39], maintaining data on the premises of a potentially vast and geographically diverse client base [45], which may encompass thousands or

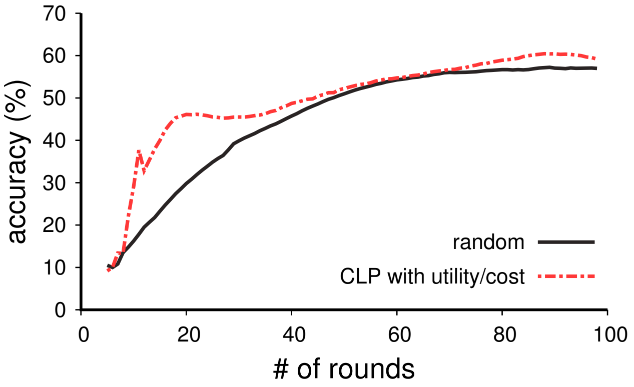


**Figure 16.** ***CIFAR10: Random vs CLP utility/cost policies. CLP saves costs while achieving the same accuracy.***

even millions of clients[21]. The pivotal distinction here lies in the fact that the data held by these clients may stem from disparate probability distributions, setting FL apart from traditional machine learning (ML) that often assumes data to be iid [24]. While prior research has delved into the non-iid aspects of FL [20, 26], our study takes a distinctive approach by exploring their interactions with variables such as cost, the statistical utility of clients, and the critical learning period.

**Client selection strategies[11, 15].** In FL, an essential challenge revolves around effectively crafting a robust model from a large pool of clients while minimizing computation and communication overhead. Numerous methodologies have emerged to enhance the employed models [6], optimize communication [3, 30, 34, 41] to and from the FL aggregator [32], and reduce the computational load borne by clients [6, 42]. Yet, a less explored terrain pertains to the identification of clients whose contributions can most significantly enhance the accuracy of the global model. Recent works have introduced a metric of statistical utility[23, 25, 28], which we leverage in CEFL. Others focused on reducing the number of clients in each round while maintaining convergence rates [7, 8]. A different perspective introduces a third criterion, exemplified by the FLAME framework [9], which assesses energy profiles for individual clients and simulates various energy profiles across the federation. However, none of these efforts have embraced a comprehensive perspective on cost, nor have they considered the impact of diverse selection strategies during critical learning periods. Additionally, certain works explored the resilience of client selection algorithms, aiming to identify and exclude malicious clients from the training process [40], while others examined incentive mechanisms to encourage clients' engagement in the federation [46], such mechanism have been explored in various multi-actor domains [14, 19, 37]. These investigations into resilience and client participation incentives complement the objectives of CEFL.

**Critical learning period** Recent studies have brought to light the significance of the initial training epochs, often referred to as the Critical Learning Period (CLP), in shaping the ultimate quality of a deep neural network (DNN) model within traditional centralized machine learning [2, 17, 22]. During this critical period, insufficient data quality or quantity in training can lead to permanent model degradation, regardless of later efforts, such as increasing the number of rounds or client epochs. This notion appears to extend to FL, as recent research[2, 3, 44] has showcased through extensive experimentation that the early learning phase significantly influences FL's final test accuracy. Some studies have even revealed the potential for heightened vulnerabilities due to adversarial attacks during the CLP [43]. CEFL seizes upon CLP insights, further expanding this concept by investigating the optimal rate of client addition or removal during and after the CLP to achieve the best performance.

**Table 3.** ***Evaluation results of CEFL's and random selection policy for additional datasets.***

| Dataset | Accuracy | Rounds to Accuracy | Cost to Accuracy (%) |
|---|---|---|---|
| **CIFAR 10** | | | |
| Random | 57.06 | 98 | 100 |
| CLP Utility/Cost | 59.53 | 72 | 57 |
| **SVHN** | | | |
| Random | 93.09 | 60 | 100 |
| CLP Utility/Cost | 93.14 | 59 | 76 |

## 7 Conclusion

In this work, we introduced CEFL, a framework that embraces the variability of costs from the client's perspective. Our definition of cost encompasses any arbitrary definition of cost from the clients' perspective including financial or environmental considerations, including CO2 emissions, exemplified throughout this study. Our evaluation revealed the inevitable trade-off between cost reduction and model accuracy. To address this challenge, we incorporated statistical utility into our client selection strategy, emphasizing utility-per-cost. Furthermore, our investigation into critical learning periods inspired dynamic client selection, optimizing cost without a significant impact on accuracy. Thorough testing across three diverse datasets and varying non-iid-ness levels underscored the versatility of CEFL. Our contributions enable cost-aware and environment friendly federated learning, addressing the needs of a broad spectrum of applications.

## References

[1] Ahmed M. Abdelmoniem, Atal Narayan Sahu, Marco Canini, and Suhaib A. Fahmy. 2023. REFL: Resource-Efficient Federated Learning. In *Proceedings of the Eighteenth European Conference on Computer Systems* (Rome, Italy) *(EuroSys '23)*. Association for Computing Machinery, New York, NY, USA, 215–232. https://doi.org/10.1145/3552326.3567485

[2] Alessandro Achille, Matteo Rovere, and Stefano Soatto. 2019. Critical Learning Periods in Deep Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, Appleton, WI.

[3] Saurabh Agarwal, Hongyi Wang, Kangwook Lee, Shivaram Venkataraman, and Dimitris Papailiopoulos. 2021. Accordion: Adaptive Gradient Communication via Critical Learning Regime Identification. In *Proceedings of Machine Learning and Systems*. mlsys.org, Indio, CA,

27 pages.

[4] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Hei Li Kwing, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390* (2020).

[5] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. mlsys.org, Indio, CA, 374–388.

[6] Sebastian Caldas, Jakub Konecˇny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *arXiv preprint arXiv:1812.07210* (2018).

[7] Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. 2018. LAG: Lazily Aggregated Gradient for Communication-Efficient Distributed Learning. In *Advances in Neural Information Processing Systems*, Vol. 31.

[8] Wenlin Chen, Samuel Horváth, and Peter Richtárik. 2022. Optimal Client Sampling for Federated Learning. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=8GvRCWKHIL

[9] Hyunsung Cho, Akhil Mathur, and Fahim Kawsar. 2022. *FLAME: Federated Learning Across Multi-Device Environments*. https://arxiv.org/abs/2202.08922 URL visited on 2023-10-04.

[10] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. 2017. EMNIST: an extension of MNIST to handwritten letters. arXiv:1702.05373 [cs.CV]

[11] Gergely Dániel Németh, Miguel Ángel Lozano, Novi Quadrianto, and Nuria Oliver. 2022. A Snapshot of the Frontiers of Client Selection in Federated Learning. *arXiv e-prints*, Article arXiv:2210.04607 (Sept. 2022), arXiv:2210.04607 pages. https://doi.org/10.48550/arXiv.2210.04607 arXiv:2210.04607 [cs.DC]

[12] European Parliament and Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council.* https://data.europa.eu/eli/reg/2016/679/oj

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local Model Poisoning Attacks to {Byzantine-Robust} Federated Learning. In *29th USENIX security symposium (USENIX Security 20)*. USENIX, Berkeley, CA, 1605–1622.

[14] Rosta Farzan, Joan M. DiMicco, David R. Millen, Casey Dugan, Werner Geyer, and Elizabeth A. Brownholtz. 2008. Results from Deploying a Participation Incentive Mechanism within the Enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 563–572. https://doi.org/10.1145/1357054.1357145

[15] 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

[16] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client-level perspective. In *NeurIPS*.

[17] Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto. 2019. Time Matters in Regularizing Deep Networks: Weight Decay and Data Augmentation Affect Early Learning Dynamics, Matter Little Near Convergence. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

[18] Google. 2023. Google Data Centers: Efficiency. http://google.com/about/datacenters/efficiency/

[19] Robert W Hahn and Robert N Stavins. 1992. Economic Incentives for Environmental Protection: Integrating Theory and Practice. *The American Economic Review* 82, 2 (1992), 464–468.

[20] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. 2020. The Non-IID Data Quagmire of Decentralized Machine Learning. In *International Conference on Machine Learning (ICML)*.

[21] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. 2022. PAPAYA: Practical, Private, and Scalable Federated Learning. In *Proceedings of Machine Learning and Systems*, D. Marculescu, Y. Chi, and C. Wu (Eds.), Vol. 4. 814–832. https://proceedings.mlsys.org/paper_files/paper/2022/file/a8bc4cb14a20f20d1f96188bd61eec87-Paper.pdf

[22] Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos J. Storkey. 2019. On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length. In *Proceedings of the International Conference on Learning Representations (ICLR)*. Appleton, WI.

[23] Tyler B. Johnson and Carlos Guestrin. 2018. Training Deep Models Faster with Robust, Approximate Importance Sampling. In *NeurIPS*.

[24] M. I. Jordan and T. M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.

[25] Angelos Katharopoulos and Francois Fleuret. 2018. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In *ICML*.

[26] Momin Ahmad Khan, Virat Shejwalkar, Amir Houmansadr, and Fatima M Anwar. 2023. On the Pitfalls of Security Evaluation of Robust Federated Learning. In *2023 IEEE Security and Privacy Workshops (SPW)*. IEEE, 57–68.

[27] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images.* Technical Report.

[28] 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, Berkeley, CA, 19–35. https://www.usenix.org/conference/osdi21/presentation/lai

[29] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. 2022. PyramidFL: A Fine-Grained Client Selection Framework for Efficient Federated Learning. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking* (Sydney, NSW, Australia) *(MobiCom '22)*. Association for Computing Machinery, New York, NY, USA, 158–171. https://doi.org/10.1145/3495243.3517017

[30] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Conference on Machine Learning and Systems (MLSys)*.

[31] Tian Li, Manzil Zaheer, Ahmad Beirami, and Virginia Smith. 2020. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations (ICLR)*. ICLR, Appleton, WI, 20 pages. https://openreview.net/forum?id=ByexElSYDr

[32] W. Y. B. Lim, N. C. Luong, D. T. Hoang, et al. 2020. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 22, 3 (2020), 2031–2063.

[33] Electricity Maps. 2022. Electricity Map. https://www.electricitymap.org/map.

[34] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, Lauderdale, FL, USA, 1273–1282.

[35] Gergely Dániel Németh, Miguel Angel Lozano, Novi Quadrianto, and Nuria M Oliver. 2022. A Snapshot of the Frontiers of Client Selection in Federated Learning. *Transactions on Machine Learning Research* (2022), 25 pages. https://openreview.net/forum?id=vwOKBldzFu Survey

Certification.

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. NeurIPS, San Diego, CA, 9 pages. http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[37] Albert L. Nichols. 1984. *Targeting Economic Incentives for Environmental Protection.* Massachusetts Institute of Technology Press, Cambridge, MA.

[38] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2022. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer* 55, 7 (2022), 18–28. https://doi.org/10.1109/MC.2022.3148714

[39] Sundar Pichai. 2019. Privacy Should Not Be a Luxury Good. https://www.nytimes.com/2019/05/07/opinion/google-sundar-pichai-privacy.html

[40] Nuria Rodríguez-Barroso, Eugenio Martínez-Cámara, M. Victoria Luzón, and Francisco Herrera. 2022. Dynamic Defense Against Byzantine Poisoning Attacks in Federated Learning. *Future Generation Computer Systems* 133 (2022), 1–9.

[41] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.

[42] Zirui Xu, Fuxun Yu, Jinjun Xiong, and Xiang Chen. 2021. Helios: Heterogeneity-Aware Federated Learning with Dynamically Balanced Collaboration. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, Piscataway, NJ, 997–1002. https://doi.org/10.1109/DAC18074.2021.9586241

[43] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. 2023. Critical Learning Periods Augmented Model Poisoning Attacks to Byzantine-Robust Federated Learning. https://openreview.net/forum?id=I7triE0okW3

[44] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. 2023. CriticalFL: A Critical Learning Periods Augmented Client Selection Framework for Efficient Federated Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 2898–2907. https://doi.org/10.1145/3580305.3599293

[45] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied Federated Learning: Improving Google Keyboard Query Suggestions. arXiv:1812.02903 [cs.LG]

[46] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. 2021. A Survey of Incentive Mechanism Design for Federated Learning. *IEEE Transactions on Emerging Topics in Computing* 10, 2 (2021), 1035–1044. https://ieeexplore.ieee.org/abstract/document/9369019

[47] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. 2020. FairFL: A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, Piscataway, NJ, 1051–1060. https://doi.org/10.1109/BigData50022.2020.9378043