Pioneering Eco-Efficiency in Cloud Computing: The Carbon-Conscious Federated Reinforcement Learning (CCFRL) Approach.

Eunil Seo, *Member, IEEE,* and Erik Elmroth, *Member, IEEE*

***Abstract*—In response to the growing emphasis on sustainability in federated learning (FL), this research introduces a dynamic, dual-objective optimization framework called Carbon-Conscious Federated Reinforcement Learning (CCFRL). By leveraging Reinforcement Learning (RL), CCFRL continuously adapts client allocation and resource usage in real-time, optimizing both carbon efficiency and model performance. Unlike static or greedy methods that prioritize short-term carbon constraints, existing approaches often suffer from either degrading model performance by excluding high-quality, energy-intensive clients or failing to adequately balance carbon emissions with long-term efficiency. CCFRL addresses these limitations by taking a more sustainable method, balancing immediate resource needs with long-term sustainability, and ensuring that energy consumption and carbon emissions are minimized without compromising model quality, even with non-IID (non-independent and identically distributed) and large-scale datasets. We overcome the shortcomings of existing methods by integrating advanced state representations, adaptive exploration and exploitation transitions, and stagnating detection using t-tests to better manage real-world data heterogeneity and complex, non-linear datasets. Extensive experiments demonstrate that CCFRL significantly reduces both energy consumption and carbon emissions while maintaining or enhancing performance. With up to a 61.78% improvement in energy conservation and a 64.23% reduction in carbon emissions, CCFRL proves the viability of aligning resource management with sustainability goals, paving the way for a more environmentally responsible future in cloud computing.**

***Index Terms*—carbon efficiency, energy efficiency, cloud computing, reinforcement learning, federated learning, non-IID data, entropy-based allocation, *t*-test, sustainable computing, green computing, decarbonization, cloud-edge continuum.**

I. Introduction

The energy supply landscape is **decentralizing**, driven by growing consumer demand and diverse power suppliers. Future power systems will comprise "BULK"—large networks for centralized Renewable Energy Sources (RES) and market integration, and "MICRO"—smaller, self-sufficient networks like microgrids with active customer involvement [1]. This transition prioritizes **decarbonization** through low-carbon energy integration and improved carbon efficiency. Our research aligns with this shift by focusing on enhancing **carbon efficiency** in **cloud computing**.

We address the challenge of **carbon emissions** in data-driven data center operations that support computing-intensive applications, such as **machine learning**, by optimizing resource allocation to balance performance with carbon reduction. Traditional data centers, which were initially designed around static grid models, are now integrating renewable energy sources like wind and solar. This integration introduces variability in both costs and emissions [2]. Our findings reveal that strategies considering carbon impacts significantly reduce emissions, highlighting the necessity for environmentally conscious approaches that align computational efficiency with sustainability.

Machine Learning (ML) is a key driver in cloud computing, leveraging infrastructure to process diverse data streams, including those from IoT devices. To optimize performance, the **cloud-edge continuum** processes data near its source [3]. Our research aims to enhance carbon efficiency in ML by exploring decentralized learning approaches like **federated learning** to reduce the carbon footprint of cloud computing.

Energy and carbon efficiency are distinct, shaped by diverse energy sources like solar, wind, and gas [4]–[7]. This research emphasizes **carbon efficiency**, prioritizing the carbon impact of energy use over mere consumption levels. Computing's adaptability allows for strategic workload scheduling to align with periods of high renewable energy availability, particularly solar and wind [8]–[11]. This approach not only reduces emissions but optimizes resource use, aligning computational efficiency with sustainability.

Given these considerations, we present the **Carbon-Conscious Federated Reinforcement Learning (CCFRL)** framework, specifically designed to optimize resource management while reducing carbon emissions in decentralized environments, such as federated learning. CCFRL leverages dynamic state representations that continuously monitor and assess key metrics—including model accuracy, model similarity, and carbon footprint—using **Reinforcement Learning (RL)**. This framework incorporates an environment-sensitive reward mechanism that balances eco-efficiency with performance, an adaptive exploration-exploitation strategy that addresses performance gains through performance stagnation detection, and an optimal client allocation method utilizing an entropy-based approach to minimize computational resource usage. These integrated features enable CCFRL to perform real-time, adaptive client allocation, ensuring resource management

Eunil Seo and Erik Elmroth are with the Department of Computing Science, Umeå University, Umeå, Sweden.
E-mail: eunil.seo@cs.umu.se, elmroth@cs.umu.se

is performance-focused and environmentally conscious. This dynamic adaptability allows CCFRL to continuously learn from and respond to changing conditions, effectively addressing performance challenges while meeting sustainability objectives.

Empirical results demonstrate that this adaptive RL approach aligns with sustainable development goals while providing a scalable, eco-efficient solution for cloud computing. CCFRL enables cloud providers to reduce emissions without compromising computational efficiency or model quality, offering a sophisticated and environmentally conscious approach essential for sustainable cloud computing. The key contributions of this research are:

1) **Introducing CCFRL**: We present **CCFRL**, a novel RL-based framework that jointly optimizes carbon efficiency and model performance in federated learning by integrating **advanced state representations**, an **environment-sensitive reward mechanism**, an **adaptive exploration-exploitation method**, and an **entropy-based client allocation strategy**.
2) **Dynamic Balance of Resource Demands and Sustainability**: CCFRL dynamically balances immediate resource demands with **long-term sustainability** by utilizing an **epsilon-greedy** RL approach and performance stagnation detection via ***t*-tests**. This ensures effective **exploration** and **exploitation**, optimal resource allocation, and avoids suboptimal client allocations, even with **non-IID (non-independent and identically distributed)** and large-scale datasets.
3) **Enhanced Management of Data Heterogeneity**: We improve the handling of **real-world data heterogeneity** and complex, non-linear datasets by leveraging the **Dirichlet distribution** to model imbalanced data and using the **Kullback-Leibler divergence** to measure differences between probability distributions, enabling more accurate management of diverse data sources.
4) **Advancing Environmentally Responsible AI**: We establish a foundation for **environmentally responsible practices** in artificial intelligence and cloud computing, demonstrating that high performance and sustainability can be achieved simultaneously.

This research addresses the rising computational demands in AI and their impact on carbon emissions, focusing on energy-efficient computing and decarbonizing cloud systems in Section II. Section III introduces a system model to enhance efficiency in federated learning by quantifying energy use and carbon emissions. Section IV presents heuristic methods—MSPCA, APA, LCA—and Randomized Double Greedy Allocation (RDGA) as baselines for benchmarking CCFRL. Section V details CCFRL's adaptive state representation, balancing performance and carbon efficiency. Section VI demonstrates significant energy and emission reductions, while Section VII highlights the need for further eco-efficient research.

II. Background and Related Work

The increasing demand for computing raises concerns about its carbon footprint [15]–[17]. AI advancements have led to energy-intensive applications, escalating costs and emissions [18]. Recent studies show that the computational demands of advanced machine learning models have been rapidly increasing [19]. Research by Beloglazov et al. [20], Guazzone et al. [21], and Tun et al. [22] highlights that energy-efficient resource management can achieve significant savings, though maintaining high performance remains a challenge. Software optimizations, as emphasized by Grosskop et al. [23], are also crucial for reducing energy consumption. While minimizing energy use is essential [4], studies by Sovacool et al. [24], Creutzig et al. [25], and Grubler et al. [26] suggest that reducing usage alone is not enough to achieve low-carbon emissions. The focus must shift towards reducing the overall carbon footprint for sustainability.

Recent research on decarbonizing data centers integrates both policy and technological strategies. Cao et al. [27] propose a roadmap for carbon-neutral data centers by increasing renewable energy use and improving energy efficiency. Ramesh et al. [28] emphasize dynamic resource management to optimize computing resource allocation, reducing energy consumption. Bergaentzle et al. [29] introduce performance indicators that go beyond conventional metrics to capture decarbonization, particularly in cooling systems. Additionally, Bashir et al. [30] and Souza et al. [31] present tools like a software-defined energy virtualization layer and the Ecovisor tool, enhancing energy control and supporting low-carbon cloud computing. This comprehensive method promotes sustainable, efficient, and decarbonized data center operations, addressing both stakeholder goals and environmental imperatives.

The research on reducing energy consumption and carbon footprints in cloud data centers focuses on various innovative approaches. Ahvar et al. [32] propose CACEV, which combines prediction-based $A*$ search and Fuzzy Sets for efficient VM placement, optimizing servers and networks. Wadhwa et al. [33] focus on VM placement and migration techniques to lower energy use and carbon emissions. Renugadevi et al. [34] introduce the C-PEF and C-FFF algorithms, which prioritize carbon-efficient clusters and power-efficient server frequencies. Khosravi et al. [35] present an ECE Cloud architecture with a heuristic VM placement algorithm to minimize carbon footprint and power consumption. Abbasi et al. [36] apply an evolutionary computing approach using a modified Memetic Algorithm to optimize VM placement, considering IoT task deadlines and energy costs. Each study contributes to enhancing the sustainability and energy efficiency of cloud data centers.

In addition to optimizing energy and carbon efficiency, the CACEV method in [32] ensures that VM allocation reduces energy and carbon footprint without compromising the IaaS Service Level Agreement (SLA), a key QoS metric. Similarly, Wadhwa et al. [33] propose the C-PEF and C-FFF algorithms, which prioritize carbon-efficient clusters and server frequencies while balancing energy conservation and QoS. Abbasi et al. [36] tackle IoT-specific challenges with a modified Memetic Algorithm, optimizing energy costs and scheduling while meeting IoT task deadlines, ensuring timely service delivery. These approaches highlight how cloud computing

TABLE I: Summary of Goals, Methods, and Limitations in Energy-Efficient Federated Learning Research.

	Energy Efficiency Goals	Energy-Efficient Method	Method Limitations
FedGreen [12], 2021	Minimize device energy consumption in FL while maintaining accuracy in MEC.	Implements fine-grained gradient compression with device-specific compression ratios and computing frequencies, and server-side gradient aggregation.	Complexity in optimizing energy-accuracy tradeoff; potential accuracy loss with high compression; added computational overhead.
GREED [13], 2022	Minimize overall energy consumption in FL by optimizing the use of renewable energy sources and prolonging device battery life.	Selects clients based on available green energy and battery levels to ensure energy-efficient participation.	Doesn't consider network variations; complex implementation; depends on renewable energy availability.
ADA [14], 2023	Minimize total energy consumption and FL time by jointly optimizing CPU frequency, bandwidth allocation, transmission power, and learning accuracy.	Utilizes the Alternative Direction Algorithm (ADA) to adjust CPU frequency, allocate bandwidth, and optimize transmission power to reduce energy consumption while meeting FL time constraints.	May increase FL time; complex implementation due to iterative optimization; relies on accurate modeling of system dynamics.

TABLE II: Overview of Goals, Methods, and Limitations in Carbon-Efficient Federated Learning Research.

	Carbon Efficiency Goals	Carbon-Efficient Approach	Approach Limitation
ECarbon [42] 2021	Quantify and reduce the carbon footprint of federated learning by analyzing energy consumption across various hardware and settings.	Develops a CO2 estimation model, considering training and communication energy, and tests aggregation strategies like FedAvg and FedAdam.	High energy consumption in non-IID settings and communication overhead, leading to inefficiencies compared to centralized learning.
FCarbon [43] 2022	Analyze the energy and carbon footprint of federated and distributed learning approaches, and determine conditions for sustainable implementations in various industrial applications.	Models energy and carbon footprints for centralized and federated learning, identifying sustainable regions based on communication efficiency, model size, and active learners.	High communication overheads in federated learning may offset carbon efficiency gains. Complexity in managing parameters like data distribution, model size, and network efficiency can limit practical deployment.

can simultaneously achieve environmental sustainability and high-quality service.

Recent research highlights the need for reducing the carbon footprint of machine learning (ML). Kaack et al. [37] and Dhar et al. [38] advocate for energy-efficient ML algorithms and hardware, along with the use of green energy in data centers. Hua et al. [39] and Yao et al. [40] emphasize ML's role in optimizing renewable energy systems, carbon markets, and smart grid management. Patterson et al. [41] highlight that ML energy efficiency varies by model type, location, and data quality, noting that sparsely activated deep neural networks are particularly efficient. Collectively, these studies call for integrating ML into energy management with careful consideration of geographical and infrastructural factors to meet sustainability goals.

The studies on energy-efficient Federated Learning (FL) aim to reduce energy consumption while maintaining performance. Li et al. [12] introduce FedGreen, employing gradient compression to reduce energy consumption in multi-access edge computing, though balancing energy efficiency and accuracy is complex and adds computational overhead. All methods face challenges related to system complexity and accuracy tradeoffs. Albelaihi et al. [13] propose selecting clients based on available renewable energy and battery levels, though this approach faces challenges such as network variability and reliance on renewable energy. Salh et al. [14] optimize CPU frequency, bandwidth, and transmission power using the Alternative Direction Algorithm, but the iterative process may extend FL time and requires precise modeling, as outlined in Table I.

Carbon-efficient Federated Learning (FL) is emerging through direct carbon footprint measurement, distinguishing carbon from energy efficiency. While earlier studies focused on improving energy efficiency for green FL, recent work highlights the importance of considering both energy use and carbon emissions [43]. Qiu et al. [42] emphasize the impact of factors like physical location, deep learning tasks, and model architecture on FL's carbon footprint, underscoring the need for dynamic resource allocation. These efforts highlight the growing focus on sustainability in advancing FL technologies, as outlined in Table II.

Directly incorporating carbon information into federated learning is becoming increasingly important for enabling carbon efficiency, particularly in optimal client selection (e.g., client allocation[1]). Several client selection/allocation methods have been developed to enhance carbon efficiency and performance in federated learning. CAFE [44] optimizes client selection based on carbon intensity and gradient contributions, using deterministic and randomized greedy approaches, though it encounters limitations in balancing performance and carbon constraints. SARIMA-FL [45] applies SARIMA-based clustering for client selection, improving carbon emissions prediction but struggling with complex datasets and client heterogeneity. D2D-FL [46] reduces energy consumption through decentralized aggregation and device-to-device communications but neglects data-driven client selection criteria. CEFL [47] selects clients based on a utility-to-cost ratio, aiming to balance carbon emissions and performance, though managing tradeoffs remains a challenge, as shown in Table III.

Our research overcomes the limitations of static and sub-

[1]Client selection and client allocation are interchangeable terms in this research. Client selection refers to determining the optimal clients for training, while client allocation focuses more on where the training occurs, considering factors such as computing power, energy consumption, and carbon intensity.

TABLE III: Comparison of Client Selection/Allocation Methods for Carbon Efficiency and Performance in Federated Learning.

	Carbon and Performance Efficiency Goals	Client Selection/Allocation Method	Method Limitations
CAFE [44], 2023	Carbon emissions reduction and performance enhancement by optimizing client selection based on carbon intensity and gradient contributions.	Deterministic Double Greedy for consistent, repeatable results; Randomized Double Greedy for exploring the solution space and avoiding local optima.	Greedy approaches often result in suboptimal performance; no comprehensive investigation of carbon constraints, leading to performance degradation.
SARIMA-FL [45], 2023	Enhance the accuracy of carbon emissions prediction and improve computational efficiency by using SARIMA-based clustering for client selection.	Clusters clients based on their seasonal and non-seasonal carbon emission trends using the SARIMA model.	SARIMA's linear modeling might not handle complex datasets; limited exploration of heterogeneous client data, leading to biased models.
D2D-FL [46], 2023	Minimize energy consumption and carbon footprint across distributed devices by leveraging device-to-device (D2D) communications.	Decentralized aggregation without a central server, overlapped clustering with bridge devices (BDs) for cross-cluster model sharing to reduce redundant transmissions.	Focus on communication efficiency without considering data-driven selection criteria like diversity and quality; dynamic selection is not explored.
CEFL [47], 2023	Minimize carbon emissions through cost-efficient client selection.	Client selection based on the utility-to-cost ratio (utility to carbon cost).	Complexity in balancing accuracy and carbon efficiency, leading to challenges in managing trade-offs.

optimal strategies in federated learning (FL) by integrating carbon efficiency through the introduction of a dynamic, dual-objective optimization framework, Carbon-Conscious Federated Reinforcement Learning (CCFRL). Unlike static or greedy approaches like CAFE [44] and CEFL [47], which often prioritize short-term carbon reduction at the expense of model performance, our RL-based framework continuously adapts to fluctuating environmental and system conditions. Existing static methods risk excluding high-quality models due to their higher energy consumption and carbon footprint, even though these models are often trained on larger, less-skewed local datasets that can significantly enhance overall performance.

In contrast, CCFRL dynamically balances immediate resource needs with long-term sustainability goals, making real-time decisions that reduce the risk of excluding models that may appear inefficient due to their higher short-term energy consumption or carbon emissions. While models trained on larger datasets or in high-carbon regions may initially contribute more emissions, they play a critical role in accelerating model convergence. CCFRL leverages an adaptive state representation and environment-involved reward mechanisms to optimize both carbon efficiency and performance, ensuring that long-term sustainability is not sacrificed for short-term gains.

Advanced features such as entropy-based client engagement and the strategic balance between exploration and exploitation, guided by performance stagnation detection, enable CCFRL to effectively manage the trade-offs between performance and carbon efficiency. These mechanisms help minimize the carbon footprint without compromising model performance.

Moreover, unlike SARIMA-FL [45], which relies on linear modeling techniques that are inadequate for handling non-linear and complex datasets, CCFRL effectively manages scalable, non-IID data distributions by employing the Dirichlet distribution and Kullback-Leibler divergence. These techniques allow CCFRL to better capture real-world heterogeneous environments. Overall, CCFRL provides a scalable, eco-efficient solution that balances sustainability with performance, addressing the limitations of earlier federated learning approaches.

III. System Models in Federated Learning for Efficiency and Sustainability

This research explores federated learning on non-IID datasets characterized by different levels of data skewness, quantified using Kullback-Leibler divergence, with a focus on improving efficiency and sustainability in model training across distributed clients. By employing the Dirichlet distribution to represent data diversity and tailoring client allocation strategies—such as Accuracy-Prioritized, Model Similarity PCA-based, and Low-Carbon Footprint—this research assesses the impact of data distribution on model performance and environmental metrics, illustrated in Fig. 1. The findings demonstrate that strategic client engagement not only enhances model accuracy but also significantly reduces carbon emissions and energy consumption, underscoring the importance of matching client allocation to data characteristics for optimizing both performance and sustainability in federated learning systems.

A. Federated learning on Kullback-Leibler (KL) divergence dataset

We explore federated learning on non-IID datasets using KL divergence, focusing on data distribution, model training, and client allocation methods to improve system efficiency and sustainability; additionally, notations used are summarized in Table IV.

1) Data Structure and Distribution Representation: This research utilizes the Dirichlet distribution, denoted as $Dir(\boldsymbol{\kappa})$, to allocate the dataset $\mathbb{D}^{\boldsymbol{\kappa}}$ across clients $\{1, \ldots, n\}$. This allocation is governed by the parameter $\boldsymbol{\kappa}$, facilitating the creation of datasets $\{\mathbb{D}_1^{\kappa_1}, \ldots, \mathbb{D}_n^{\kappa_n}\}$ that are distributed in a manner mirroring realistic scenarios. Here, κ_n for $n = 1, \ldots, N$ signifies the parameter that quantifies the diversity inherent in the dataset dispensed to the n-th client. Additionally, κ_n is employed as a metric to gauge the congruence between the data distribution of each client and the anticipated overall distribution. To assess the diversity of the dataset associated with each client, denoted as κ_n, this research advocates the use

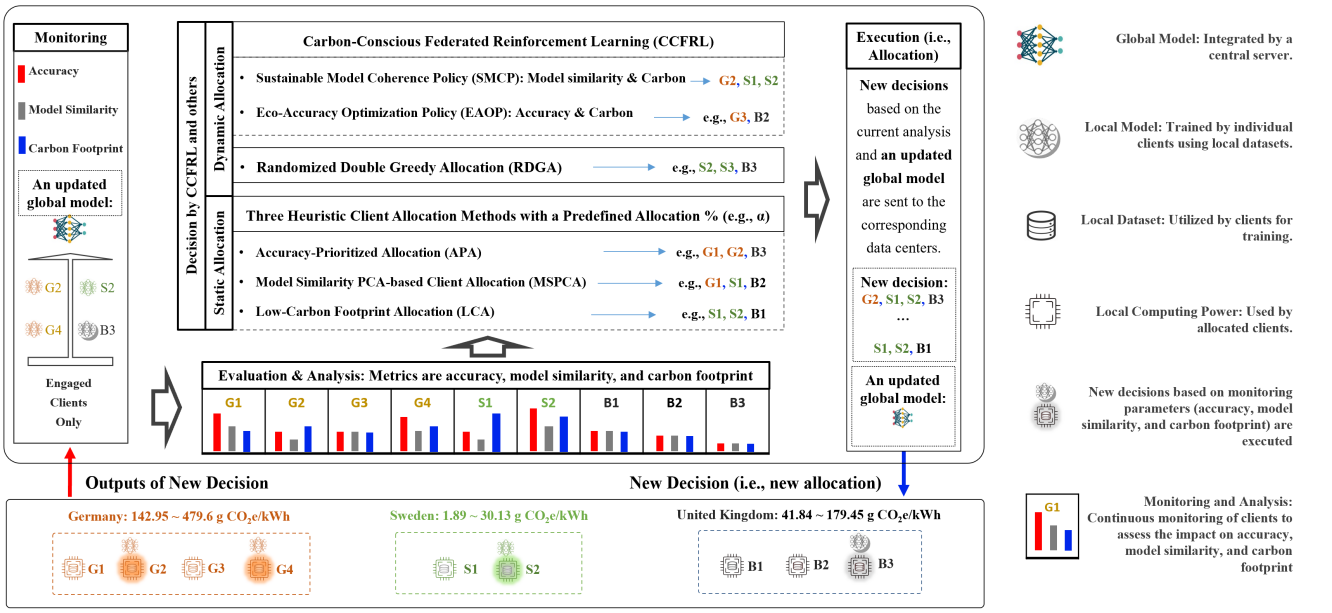


Fig. 1: The proposed Carbon-Conscious Federated Reinforcement Learning (CCFRL) framework dynamically allocates clients by optimizing both model quality (accuracy and model similarity) and carbon footprint. CCFRL integrates two strategies: Eco-Accuracy Optimization and Sustainable Model Coherence, both of which significantly improve energy and carbon efficiency while maintaining high model performance. Additionally, the Three Heuristic Client Allocation Strategies with Static Allocation and the Randomized Double Greedy Allocation are used as benchmarks for comparison. This framework continuously monitors, evaluates, and adjusts decisions in real-time, ensuring a balanced and sustainable approach to federated learning.

TABLE IV: Notations in FL on KL divergence dataset.

Notation	Description
$\mathbb{D}^{\boldsymbol{\kappa}}$	Distributed dataset by Dirichlet distribution $\boldsymbol{\kappa}$
κ_n	KL divergence value for client n, resulted from $\boldsymbol{\kappa}$
N	Number of clients
$\mathbb{D}_n^{\kappa_n}$	Local dataset for client n by κ_n
$\mathbb{X}_i^d$	Input features for sample i
y_i	Output label for sample i
D_n	Number of samples for client n
$D_{KL}(Q\|P)$	KL divergence between distributions Q and P
P	Aggregate dataset distribution
Q	Local dataset distribution
$\boldsymbol{\omega}$	Global model parameters
$\boldsymbol{\omega}_n$	Local model parameters for client n
F_n	Local loss function for client n
C	Number of classes
$\delta_{\mathbb{D}^{\boldsymbol{\kappa}},D}$	Maximum accuracy for dataset
L	Number of communication rounds
ψ	Utility function
β_1, β_2	Utility function parameters
L_{opt}	Optimal communication rounds for benchmark
$Q_n^{\mathcal{S}}$	Quality metric for allocated client n
$Q_n^{\mathcal{NS}}$	Quality metric for non-allocated client n
$\boldsymbol{\alpha}$	Proportion to allocate client, e.g., 50%, ..., 90%
$\mathcal{S}$	Set of allocated clients
$\mathcal{NS}$	Set of non-allocated clients
$\mathbb{D}_{proxy}$	Proxy dataset on the server
δ	Factor for adjusting quality metric
θ	Base adjustment for quality score
ϵ	Small constant to prevent division by zero
Cb_n	Carbon footprint for client n

of statistical diversity indices or distribution similarity metrics, such as the Kullback-Leibler (KL) divergence. This approach enables a comparative analysis of each client's data distribution against the expected overall distribution, offering insights into the extent of diversity and distributional similarity.

To analyze the statistical diversity of local datasets, we examine the structure of the local data, denoted by $\mathbb{D}_n^{\kappa_n}$, which is composed of the input-output pairs $\left\{(\mathbb{X}_i^d, y_i)\right\}_{i=1}^{D_n}$. Here, $\mathbb{X}_i^d$ represents the input data, consisting of a set of features $\left\{x_{i1}, x_{i2}, \ldots, x_{id}\right\}$, where each x_{ij} is a feature of the input data, with d indicating the total number of features, such as the 1024 pixels in a single image from the CINIC10 dataset. The term y_i corresponds to the labeled output value associated with $\mathbb{X}_i^d$, and D_n specifies the total number of data samples in the dataset.

For discrete probability distributions, the Kullback-Leibler (KL) divergence, denoted as κ_n, between an aggregate dataset distribution P and a local dataset distribution Q is quantified by summing over all possible labels l. This relationship is mathematically represented as:

$$\kappa_n = D_{KL}(Q\|P) = \sum_l Q(y=l) \log\left(\frac{Q(y=l)}{P(y=l)}\right) \quad (1)$$

where $Q(y=l)$ and $P(y=l)$ denote the probabilities of observing a data sample with label l in the local dataset distribution Q and the aggregate dataset distribution P, respectively.

The KL divergence, $D_{KL}(Q\|P)$, spans from 0 to infinity, as indicated by the expression:

$$D_{KL}(Q\|P) \in [0, \infty) \quad (2)$$

The interpretation of KL divergence values is as follows:

- A KL divergence of 0 implies perfect identity between the two distributions, where $D_{KL}(Q\|P) = 0$ holds if and only if $Q(y=l) = P(y=l)$ for every label l.

- Values greater than 0 indicate a discrepancy between the distributions. The larger the KL divergence $\boldsymbol{\kappa}$, the more significant the difference between the two distributions, reflecting varying degrees of distributional disparity.

2) Learning on non-IID datasets distributed by $\boldsymbol{\kappa}$: We present the framework of federated learning with heterogeneous data. We first introduce the FL training goal and quantify data heterogeneity. We then show the model update process.

We consider a server that initiates a federated learning process and selects clients from the client set $\mathcal{N}$ ($|\mathcal{N}| = N$) to train a global model $\boldsymbol{\omega}$ represented by a parameter vector. Each client $n \in \mathcal{N}$ has a local dataset with $\mathbb{D}_n^{\kappa_n}$ local data samples, and clients have different data distributions κ_n. The server selects a subset $\mathcal{S} \subseteq \mathcal{N}$ of clients for model training. The training goal of federated learning is to obtain the optimal global model, which can minimize the global loss function.

$$F(\boldsymbol{\omega}) = \sum_{n \in \mathcal{S}} \frac{D_n}{\sum_{n' \in \mathcal{S}} D_{n'}} F_n(\boldsymbol{\omega}; \mathbb{D}_n^{\kappa_n}), \tag{3}$$

where the weight $\frac{D_n}{\sum_{n' \in \mathcal{S}} D_{n'}}$ reflects the relative size of client n's dataset within the allocated set $\mathcal{S}$, ensuring fair contribution to the global model update. The function F: $\mathbb{R}^d \to \mathbb{R}, n = 1, \ldots, S$, signifies the expected loss over the distribution of the local dataset $\mathbb{D}_n^{\kappa_n}$ for each device n.

In this model, F_n symbolizes the expected loss for the n-th client's dataset, mapped from the space of parameters $\mathbb{R}^d$ to real numbers, reflecting the anticipated loss over the local dataset's distribution $\mathbb{D}_n^{\kappa_n}$. This formulation integrates the dataset diversity represented by κ_n, with the possibility of employing statistical metrics like the Kullback-Leibler divergence to quantify and compare the diversity of these local distributions.

We also examine the specifics of the data and loss calculation for each client. Each client n possesses a local dataset $\mathbb{D}_n^{\kappa_n}$ with D_n denoting the number of samples therein. The true label for the i-th sample in class C is represented by y_{ic}, typically in a one-hot encoded vector format. The predicted probability that the i-th sample belongs to class C is denoted by $\hat{y}_{ic}(\boldsymbol{\omega}_n)$, as determined by the model with parameters $\boldsymbol{\omega}_n$.

In this federated learning setup, the local loss function for client n, considering the cross-entropy loss across its dataset, is given by:

$$F_n(\boldsymbol{\omega}_n^t; \mathbb{D}_n^{\kappa_n}) = -\sum_{i=1}^{D_n} \sum_{c=1}^{C} y_{ic} \log(\hat{y}_{ic}(\boldsymbol{\omega}_n^t;)). \tag{4}$$

This cross-entropy loss function aggregates the individual losses from each data sample, capturing the discrepancy between the actual labels and the predicted class probabilities across all classes with respect to the diversity and distribution of data across clients as encoded by the Dirichlet distribution parameters κ_n.

The stochastic gradient descent (SGD) algorithm updates the global model parameters $\boldsymbol{\omega}$ iteratively based on gradients computed from a subset of the allocated clients $\mathcal{S}$. At each iteration t, the update equation for the global model parameters $\boldsymbol{\omega}$ is:

$$\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \eta \sum_{n \in \mathcal{S}} \frac{D_n}{\sum_{n' \in \mathcal{S}} D_{n'}} \nabla F_n(\boldsymbol{\omega}_n^t; \mathbb{D}_n^{\kappa_n}), \tag{5}$$

where η is the learning rate and $\nabla F_n(\boldsymbol{\omega}_n^t; \mathbb{D}_n^{\kappa_n})$ represents the gradient of the local loss function F_n with respect to the parameters $\boldsymbol{\omega}$ computed on client n's dataset $\mathbb{D}_n^{\kappa_n}$ at iteration t.

3) Utility of a non-IID dataset: In this section, we examine the intricacies of non-IID (not independently and identically distributed) datasets, drawing on the foundational equation presented in Armstrong et al. [48]. The efficacy of the federated learning model $\boldsymbol{\omega}$ is pivotal and primarily dependent on two factors: the dataset size D and the label distribution $\boldsymbol{\kappa}$. These elements are seamlessly integrated into the optimization framework, expressed mathematically as:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega} \in \mathbb{R}^d} F(\boldsymbol{\omega}; \mathbb{D}^{\boldsymbol{\kappa}, D})$$

Here, $\boldsymbol{\omega}^*$ symbolizes the optimized model parameters, derived from the entire non-IID dataset $\mathbb{D}^{\boldsymbol{\kappa}}$, with its effectiveness being influenced by both the size of the dataset D and the label distribution $\boldsymbol{\kappa}$.

To quantify the optimal performance capability of $\mathbb{D}^{\boldsymbol{\kappa}, D}$, we introduce the metric $\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}}$:

$$\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}} = \Delta(\boldsymbol{\kappa}, D),$$

where Δ signifies a non-decreasing function that projects the maximum attainable accuracy, accounting for the size of the dataset D and the mean label distribution $\boldsymbol{\kappa}$ across all clients.

Leveraging insights from Eq. 6 in [49], the utility of the non-IID dataset $\mathbb{D}^{\boldsymbol{\kappa}, D}$ is defined in relation to the number of communication rounds L between the server and the clients, which is crucial for federated learning efficacy.

The utility function, $U_{\mathbb{D}^{\boldsymbol{\kappa}, D}}$, is represented as:

$$U_{\mathbb{D}^{\boldsymbol{\kappa}, D}} = \psi(L; \beta = [\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}}, \beta_1, \beta_2]) = \delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}} \cdot \left(1 - \beta_1 e^{-\beta_2 \cdot L}\right),$$

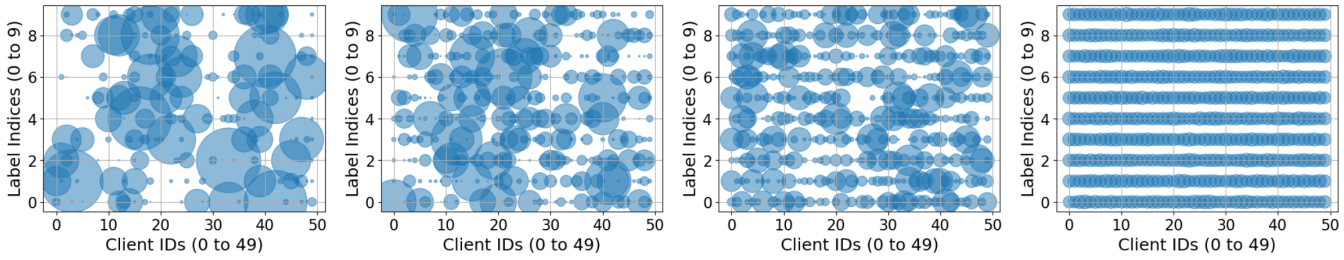
where ψ is a non-decreasing function depicting the concept of diminishing returns on accuracy improvements, with β_1 and β_2 as the parameters defining the function's progression and its approach to the limit $\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}}$.

To determine the optimal number of global iterations, L_{opt}, for achieving a predefined quality benchmark Q_g, the following relationship is used:

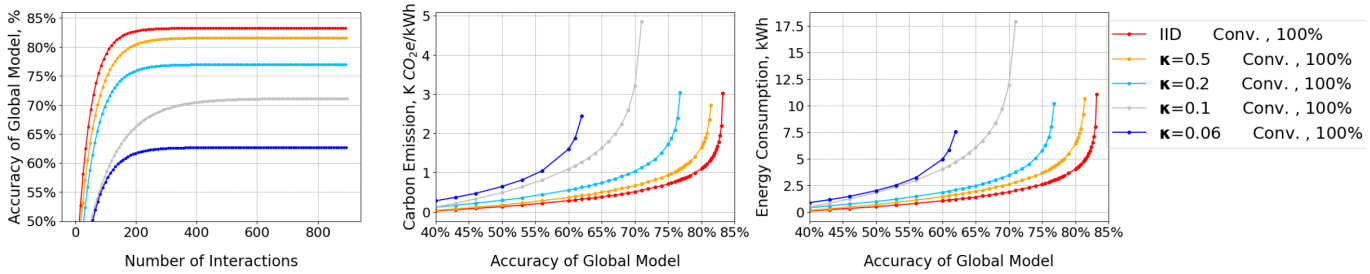
$$L_{opt}^{Q_g} \geq \frac{\log\left(\frac{\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}} \cdot \beta_1}{\delta_{\mathbb{D}^{\boldsymbol{\kappa}, D}} - Q_g}\right)}{\beta_2}, \tag{6}$$

where $L_{opt}^{Q_g}$ represents the minimum number of communication rounds required to attain or exceed the quality benchmark Q_g in the federated learning environment, ensuring a harmonized and efficient training across the networked, diverse datasets.

4) Evaluating Performance Across Varied Data Distributions: Given the foundation laid in the previous sections, we thoroughly evaluate how diverse data distributions impact the efficiency of training models. In this section, we meticulously analyzed the influence of the Dirichlet distribution parameter $\boldsymbol{\kappa}$ on data distribution, as illustrated in Fig. 2.



(a) Visualization of the impact of different Dirichlet distribution parameters ($\boldsymbol{\kappa}$ = (0.06, 0.1, 0.5, 100), with $\boldsymbol{\kappa} = 100$ treated as $\boldsymbol{\kappa} = \infty$ or IID) on the distribution of labels across clients. Each panel presents a unique subdivision of dataset D into sub-datasets $(D_1^{\kappa_1}, \ldots, D_n^{\kappa_n})$ where κ_n values vary according to $\boldsymbol{\kappa}$, highlighting the impact of label skewness on data availability and potential learning challenges.



(b) Comparative analysis of model performance across datasets with varying $\boldsymbol{\kappa}$ values, trained using a conventional approach without client-specific allocation. This panel demonstrates that less skewed distributions (closer to IID) yield higher accuracies and lower environmental impacts, while highly skewed datasets ($\boldsymbol{\kappa} = 0.06$) lead to reduced model performance and increased environmental impacts.

Fig. 2: Exploration of model efficacy and environmental considerations in relation to dataset label skewness across various $\boldsymbol{\kappa}$ values, highlighting the significant influence of data distribution on performance metrics in machine learning deployments.

a) Experimental Setup: We employed the CINIC-10 dataset, an extended version of CIFAR-10, combining elements from ImageNet to create a challenging benchmark. The datasets encompass a total of 270,000 and 60,000 32x32 pixel images, respectively, across 10 classes, as cited in [50], [51]. We configured experiments to cover a spectrum of skewness values denoted by $\boldsymbol{\kappa} = (0.06, 0.1, 0.5, 100 \equiv \infty)$, reflecting both IID and non-IID scenarios. Hereafter, $\boldsymbol{\kappa} = 100$ will indicate $\boldsymbol{\kappa} = \infty$.

b) Data Analysis: Figure 2a illustrates the variation in label distribution across clients for different $\boldsymbol{\kappa}$ parameters, comparing two extremes. At one end, $\boldsymbol{\kappa} = \infty$ represents a homogeneous distribution, ensuring uniform label presence across all clients. Conversely, $\boldsymbol{\kappa} = 0.06$ exhibits high skewness, leading to significant label imbalances and data scarcity for several labels within many clients, thus confirming theoretical predictions about the impacts of diversity based on Dirichlet distribution parameters.

c) Dataset Partitioning: The dataset $\mathbb{D}$ is partitioned into sub-datasets $\mathbb{D}_n^{\kappa_n}$, influenced by the Dirichlet distribution parameter $\boldsymbol{\kappa}$. This parameter modulates the diversity and similarity of data across various clients, as detailed below and Table V:

- $\boldsymbol{\kappa} = 0.06$: The mean is 13.187 and the standard deviation is 3.406, with κ_n values from 0.353 to 18.42, indicating high diversity and significant variation among clients.
- $\boldsymbol{\kappa} = 0.1$: Characterized by a mean of 10.76 and a standard deviation of 3.287, and κ_n values from 0.86 to 18.42, reflecting substantial heterogeneity but a reduced variation compared to $\boldsymbol{\kappa} = 0.06$.
- $\boldsymbol{\kappa} = 0.2$: With a mean of 6.582 and a standard deviation of 2.778, and κ_n values from 0.097 to 12.66, this setting indicates moderate diversity.
- $\boldsymbol{\kappa} = 0.5$: Presents a mean of 1.91 and a standard deviation of 1.355, with κ_n values from 0.203 to 5.243, leading to a more homogeneous distribution.
- $\boldsymbol{\kappa} = \infty$: Results in a mean of 0.0034 and a standard deviation of 0.0016, with κ_n values from 0.0097 to 0.0014, showcasing minimal diversity and almost IID condition.

TABLE V: Influence of the Dirichlet distribution parameter $\boldsymbol{\kappa}$ on data diversity among clients. Hereafter, $\boldsymbol{\kappa} = 100$ will indicate $\boldsymbol{\kappa} = \infty$.

$\boldsymbol{\kappa}$	Mean	Std Dev	κ_n Range	Description
0.06	13.19	3.41	0.354 to 18.42	High diversity, significant variation
0.1	10.77	3.29	0.86 to 18.42	Substantial heterogeneity
0.5	1.91	1.36	0.204 to 5.24	More homogeneous distribution
100	3.5e-3	1.6e-3	9.7e-3 to 1.4e-3	Minimal diversity, nearly IID condition

TABLE VI: Performance metrics across $\boldsymbol{\kappa}$ settings, showing maximum accuracy, CO_2 emissions, and energy consumption for 60%, 70%, and 80% accuracy. Lower $\boldsymbol{\kappa}$ values increase emissions and consumption due to higher data non-IIDness. NaN indicates that the accuracy threshold was not achieved under those conditions.

$\boldsymbol{\kappa}$	Maximum Acc. (%)	For achieving 60% Acc.		For achieving 70% Acc.		For achieving 80% Acc.	
		Emission (CO_2 kg)	Consumption (kWh)	Emission (CO_2 kg)	Consum. (kWh)	Emission	Consumption
∞	83.18	0.28	1.04	0.51	1.87	1.10	4.02
0.5	81.57 (-1.61%)	0.37 (+28.5%)	1.42 (+36.5%)	0.67 (+30.7%)	2.60 (+38.6%)	1.65 (+49.9%)	6.41 (+59.4%)
0.2	74.32 (-8.86%)	0.55 (+94.8%)	1.84 (+77.0%)	1.04 (+102.9%)	3.45 (+84.4%)	NaN	NaN
0.1	71.12 (-12.06%)	1.09 (+284.0%)	4.02 (+286.1%)	3.21 (+527.9%)	11.95 (+638.3%)	NaN	NaN
0.06	62.68 (-20.5%)	1.60 (+462.7%)	4.95 (+376.1%)	NaN	NaN	NaN	NaN

This distribution representation emphasizes the utility of $\boldsymbol{\kappa}$ as a tunable parameter for managing data heterogeneity among clients in distributed learning environments. The diverse κ_n values across different $\boldsymbol{\kappa}$ settings serve as a practical example of how data distribution can be strategically diversified or homogenized for machine learning applications.

d) Performance Metrics: We assess model performance by analyzing accuracy, environmental impact (measured by carbon emissions and energy consumption), and model convergence speed. The relationship between $\boldsymbol{\kappa}$ values and these metrics is critical, as shown in Figure 2b, where models trained on less skewed data distributions generally outperform those trained on more skewed datasets.

e) Statistical Evaluation: We conducted an in-depth analysis utilizing the Kullback-Leibler (KL) divergence to gauge the disparity between local and global data distributions. Our statistical findings revealed a clear correlation: higher KL divergence values, indicative of greater data distribution non-IIDness, align with declines in model performance. These trends are systematically quantified across varying $\boldsymbol{\kappa}$ settings as shown in the integrated metrics table VI.

For instance, at $\boldsymbol{\kappa} = 0.06$, where skewness is maximized, the model's accuracy significantly decreases to 62.68%, and resource consumption in terms of energy (4.95 kWh) and carbon emissions (1.60 kg CO_2) is notably high, underlining the operational inefficiencies tied to severe data skewness. In contrast, at $\boldsymbol{\kappa} = \infty$, representing ideal homogeneous data distribution, the model achieves a higher accuracy of 83.18% with lower emission and consumption rates at higher accuracies.

Multiple experimental iterations, indexed by different $\boldsymbol{\kappa}$ values, were performed to ensure the statistical robustness of these findings. This rigorous validation confirms consistent performance trends across setups, further substantiating that the degree of non-IIDness in data distributions critically affects outcomes in federated learning systems. These insights are crucial for optimizing federated learning deployments in practical applications, emphasizing the need to manage data heterogeneity strategically.

B. Dynamic Quantification of Energy Consumption and Carbon Emission

We propose models for energy consumption and carbon emissions. The notations used are summarized in Table VII.

1) Energy Consumption in GPU-Based Distributed Computing: This research extends the model presented in Eq. 11 of [52], which calculates the energy consumption in CPU-based computing environments. Our revised models are specifically tailored for GPU-based distributed computing systems. These models account for factors like actual running times, average power consumption, and the dynamics of communication and client participation in GPU-based systems.

TABLE VII: Notations used to define the system model.

Notation	Description
E_{run}	Energy consumed during a GPU run
P_{avg}	Average power consumption of the GPU
T_{run}	Task running time on the GPU
E_{total}	Total energy consumption across all clients' GPUs
$P_{avg,n}$	Average power consumption of each client's GPU
$T_{run,n}$	Running time of each client's task on the GPU
$E_{allocate}$	Energy consumption with client allocation probability
$\delta_{n,\gamma}$	Binary function, the allocation of the n^{th} client
E_{comm}	Energy consumption for communication
M	Model size
r_{avg}	Average power consumption of the comm. hardware
S_{avg}	Average network speed
$CO_2e^{n,t}$	Carbon intensity for client n at time t
$CO2e_{\text{total}}$	Total carbon emissions for all clients
$CO2e_{\text{allocate}}$	Carbon emissions for allocated clients
$CO2e_{\text{comm}}$	Carbon emissions attributed to communication
$L_{opt}^{Q_g}$	Optimal training number for a given quality goal Q_g

The energy consumption for GPU processing during a run is defined as follows:

$$E_{run} = P_{avg} \cdot T_{run} \tag{7}$$

where E_{run} is the energy consumed during the run, P_{avg} is the average power consumption of the GPU, and T_{run} is the task's running time.

To model the energy consumption in the context of training a global model that integrates N local models from N clients in a single global iteration, we define:

$$E_{total} = \sum_{n=1}^{N} P_{avg,n} \cdot T_{run,n} \tag{8}$$

where E_{total} is the total energy consumption across all GPUs, N is the number of clients (GPUs), $P_{avg,n}$ is the average power consumption of each GPU, and $T_{run,n}$ is the running time for each GPU.

To further refine the model considering the client allocation probability, we define the energy consumption as:

$$E_{allocate} = \sum_{n=1}^{N} P_{avg,n} \cdot T_{run,n} \cdot \delta_{n,\gamma} \tag{9}$$

where $E_{allocate,GPU}$ denotes the energy consumption factoring in the client allocation probability, $\delta_{n,\gamma}$ is a binary function representing the allocation of the n^{th} client.

Finally, the energy consumption attributed to communication, especially during the transmission of model parameters between the server and clients, is defined as:

$$E_{comm} = \frac{M \cdot r_{avg}}{S_{avg}} \tag{10}$$

where $E_{comm,GPU}$ is the energy consumption for communication in the GPU-based system, M is the model size, r_{avg} is the average power consumption of the communication hardware during transmission, and S_{avg} represents the average network speed.

2) Dynamic Carbon Emission Quantification: To accurately measure carbon emissions from energy use during training, we consider the carbon intensity (CO_2e), representing the CO_2 emissions per kilowatt-hour of electricity consumed. This rate varies internationally, reflecting the differing energy sources. Our analysis of distributed learning examines how carbon emissions fluctuate with the allocation of clients from various geographic locations, each with distinct CO_2e levels.

Total carbon emissions for all clients are calculated as follows:

$$CO_2e_{\text{total}} = \sum_{n=1}^{N} \left(\sum_{t=1}^{T} CO_2e^{n,t} \cdot P_{\text{avg},n} \cdot T_{n,t} \right), \tag{11}$$

where N is the total number of clients, $CO_2e^{n,t}$ is the carbon intensity for client n at time t, and $T_{n,t}$ is the training time.

For the allocated clients, considering their energy efficiency and clean energy utilization, carbon emissions are:

$$CO2e_{\text{allocate}} = \sum_{n=1}^{S} \left(\sum_{t=1}^{T} CO_2e^{n,t} \cdot P_{\text{avg},n} \cdot T_{n,t} \right), \tag{12}$$

with S representing the subset of allocated clients, facilitating targeted reductions in emissions by prioritizing clients with lower CO_2e.

For carbon emissions considering client allocation probability in a single global iteration, we define:

$$\begin{aligned} CO2e_{\text{allocate}} &= CO_2e^{n} \cdot E_{\text{allocate, GPU}}, \\ &= \sum_{n=1}^{S} CO_2e^{n} \cdot P_{\text{avg},n} \cdot T_{n}. \end{aligned} \tag{13}$$

And for carbon emissions attributed to communication in a single global iteration, we define:

$$\begin{aligned} CO2e_{\text{comm}} &= CO_2e^{n} \cdot E_{\text{comm, GPU}}, \\ &= \sum_{n=1}^{S} CO_2e^{n} \cdot P_{\text{avg},n} \cdot \frac{M \cdot r_{\text{avg}}}{S_{\text{avg}}}. \end{aligned} \tag{14}$$

To determine the optimal carbon emissions for achieving a specific quality goal Q_g, the emissions for both allocation and communication over the optimal number of training iterations $L_{\text{opt}}^{Q_g}$ are summed up:

$$CO2e_{\text{allocate}}^{Q_g} = L_{\text{opt}}^{Q_g} \cdot \sum_{n=1}^{S} CO_2e^{n} \cdot P_{\text{avg},n} \cdot \left(T_{\text{run},n} + \frac{M \cdot r_{\text{avg}}}{S_{\text{avg}}} \right), \tag{15}$$

highlighting the environmental impact of achieving the desired quality in distributed learning systems.

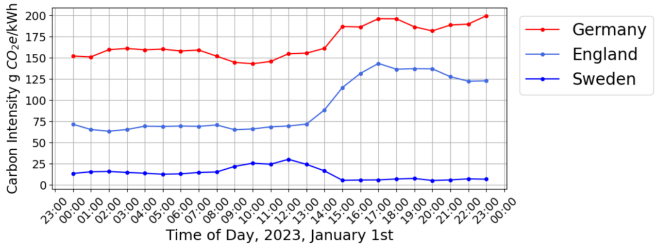


Fig. 3: Comparison of daily CO_2e carbon intensity across Germany, the United Kingdom, and Sweden, which illustrates regional disparities, highlighting the need for tailored carbon management strategies.

IV. Three Heuristic Client Allocation Methods and Randomized Double Greedy Method

This research presents three heuristic client allocation methods—Model Similarity PCA-based Client Allocation (MSPCA), Accuracy-Prioritized Allocation (APA), and Low-Carbon Footprint Allocation (LCA)—along with the Randomized Double Greedy Allocation (RDGA) as baselines for comparison. These methods serve as benchmarks to evaluate the effectiveness of the proposed Carbon-Conscious Federated Reinforcement Learning (CCFRL) method. By comparing CCFRL with these existing methods, we aim to demonstrate its feasibility and effectiveness in optimizing performance and carbon efficiency in federated learning environments.

A. Model Similarity PCA-based Client Allocation (MSPCA)

The Model Similarity PCA-based Client Allocation (MSPCA) leverages the similarity among local models to guide client allocation. After each client n trains its local model $\boldsymbol{\omega}_n$ on its dataset $\mathbb{D}_n^{\kappa_n}$, the resulting high-dimensional model is sent to the server. The server then applies Principal Component Analysis (PCA) to project all local models into a lower-dimensional space, capturing the most relevant features for comparison. The similarity between models is quantified by computing the Euclidean distance between their projections, with smaller distances indicating higher similarity. Using these distances, the server allocates clients based on model similarity, ensuring that the chosen clients contribute optimally to the overall learning objective, such as improving model convergence or maintaining diversity.

The quality metric $Q_n^{\mathcal{S}}$ for selecting clients is inversely related to the PCA distance, where a shorter distance indicates a higher similarity to the collective feature space of all local models. It is defined as:

$$Q_n^{\mathcal{S}} = \frac{1}{d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{center}) + \epsilon},$$

where ϵ is a small constant to prevent division by zero. Here, $\boldsymbol{\omega}_n$ represents the PCA-transformed weights of the client n, and $\boldsymbol{\omega}_{center}$ denotes the centroid of the PCA-transformed weights from all local models.

For clients not allocated in the current round ($n \in \mathcal{NS}$), their quality score $Q_n^{\mathcal{NS}}$ is updated using

$$Q_n^{\mathcal{NS}} = Q_{avg}^{\mathcal{S}} \cdot (\text{random}() \cdot \delta + \theta),$$

to ensure fairness and give every client a chance to contribute over time. The server iterates through these steps to evaluate the PCA-based distance, allocate clients based on the quality metric, and update the quality scores. This method aims to enhance the global model's performance while ensuring fairness and diversity in client participation, focusing on the alignment between local and global models in the feature space in Algorithm 1.

Algorithm 1 Model Similarity PCA-based Client Allocation (MSPCA) in Federated Learning.

Require:

Refer to Shared Initial Conditions for Fl Algorithms:
N: Total number of clients
$\boldsymbol{\alpha}$: Proportion to allocate client, e.g., 50%, ..., 90%
$\mathcal{S}$: Set of allocated clients
$\mathcal{NS}$: Set of non-allocated clients
$\mathbb{D}_{proxy}$: Proxy dataset on the server
δ: Factor for adjusting quality metric, $0 < \delta \leq 0.2$
θ: Base adjustment for quality score, $0.9 \leq \theta < 1$
while not converged **do**
  if first round **then**
    $\mathcal{S} \leftarrow 1, 2, \ldots, N$, $\mathcal{NS} \leftarrow \emptyset$
  else
    Proportionally determine $\mathcal{S}$ and $\mathcal{NS}$ from all clients based on the updated quality metrics $Q_n^{\mathcal{S}}$ & $Q_n^{\mathcal{NS}}$ using $\boldsymbol{\alpha}$.
  end if
  for each client $n \in \mathcal{S}$ **do**
    Train local model $\boldsymbol{\omega}_n$ on local dataset $\mathbb{D}_n^{\kappa_n}$
    Project all client models' weights using PCA
    Compute the centroid $\boldsymbol{\omega}_{\text{center}}$ of PCA-transformed weights
    Project $\boldsymbol{\omega}_n$ using PCA
    Compute Euclidean distance $d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{\text{center}})$
    Calculate quality metric: $Q_n^{\mathcal{S}} = \frac{1}{d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{\text{center}}) + \epsilon}$
  end for
  Update global model $\boldsymbol{\omega}$ using models from $\mathcal{S}$
  Calculate $Q_{avg}^{\mathcal{S}}$ from $Q_n^{\mathcal{S}}$ for all $n \in \mathcal{S}$
  for each client $n \in \mathcal{NS}$ **do**
    Update $Q_n^{\mathcal{NS}} = Q_{avg}^{\mathcal{S}} \cdot (\text{random}() \cdot \delta + \theta)$
  end for
end while
Output the improved global model

B. Accuracy-Prioritized Allocation (APA)

Accuracy-Prioritized Allocation (APA) evaluates each client's local model using a proxy dataset stored on the server, providing a standardized and direct measure of performance. Unlike MSPCA, which relies on model similarity, APA assesses model performance based on metrics such as accuracy, precision, and recall derived from a proxy dataset $\mathbb{D}_{\text{proxy}}$.

After each client n trains its local model $\boldsymbol{\omega}_n$ on its local dataset $\mathbb{D}_n^{\kappa_n}$, the server evaluates the model on the proxy dataset:

$$\text{Eval}(\boldsymbol{\omega}_n; \mathbb{D}_{\text{proxy}}),$$

providing a uniform benchmark for comparison across clients. The quality metric $Q_n^{\mathcal{S}}$ in APA is a function of both the local model's gradient information and its performance on the proxy dataset:

$$Q_n^{\mathcal{S}} = f\left(\nabla F_n(\boldsymbol{\omega}_n; \mathbb{D}_n^{\kappa_n}), \text{Eval}(\boldsymbol{\omega}_n; \mathbb{D}_{\text{proxy}})\right).$$

This ensures that both the learning progress and the immediate performance of the model are taken into account when determining client allocation.

While APA provides an objective way to measure client contributions, it has notable drawbacks. The use of a proxy dataset imposes additional computational overhead and is often impractical in many Federated Learning (FL) scenarios due to concerns over data privacy, security, and the difficulty of maintaining a representative dataset that reflects the diverse local environments of the clients.

To ensure fairness, clients that are not selected in a given round have their quality scores updated using the mechanism described in MSPCA, ensuring all clients have a chance to be chosen over time.

Algorithm 2 Accuracy-Prioritized Allocation in FL.

Require:

Refer to Shared Initial Conditions for FL Algorithms
while not converged **do**
  Follow the client allocation step as in Algorithm 1
  for each client $n \in \mathcal{S}$ **do**
    Train local model $\boldsymbol{\omega}_n$ on local dataset $\mathbb{D}_n^{\kappa_n}$
    Compute gradient $\nabla F_n(\boldsymbol{\omega}_n; \mathbb{D}_n^{\kappa_n})$
    Evaluate $\boldsymbol{\omega}_n$ using $\text{Eval}(\boldsymbol{\omega}_n; \mathbb{D}_{\text{proxy}})$
    Calculate quality metric: $Q_n^{\mathcal{S}} = f\left(\nabla F_n(\boldsymbol{\omega}_n; \mathbb{D}_n^{\kappa_n}), \text{Eval}(\boldsymbol{\omega}_n; \mathbb{D}_{\text{proxy}})\right)$
  end for
  Update global model and client quality scores following the methods in Algorithm 1
end while
Output the improved global model

APA iterates through these steps, selecting clients based on performance and learning progress. Although it offers an objective comparison through a proxy dataset, it comes with additional computational, potentially making it less practical due to data privacy, as shown in Algorithm 2.

C. Low-Carbon Footprint Allocation (LCA)

In this Carbon-Conscious Client Allocation Method, the primary objective is to minimize the carbon footprint of the federated learning process while maintaining model performance.

Each client n assesses its carbon footprint Cb_n, which reflects the energy consumption and efficiency of the hardware used during local model training $\boldsymbol{\omega}_n$ on its dataset $\mathbb{D}_n^{\kappa_n}$.

The carbon footprint is a key factor in guiding the client selection process, aiming to favor clients with more sustainable practices.

The quality metric $Q_n^{\mathcal{S}}$ for selecting clients combines both the model similarity and their environmental impact. It is defined as:

$$Q_n^{\mathcal{S}} = \frac{1}{d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{center}) \times \eta_n + \epsilon},$$

where $d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{center})$ represents the PCA distance between the PCA-transformed weights of the client n and the centroid of the PCA-transformed weights from all local models, and η_n is the carbon-based incentive inversely related to the client's carbon footprint Cb_n. Specifically, $\eta_n = \frac{k}{Cb_n}$, where k is a scaling constant.

In this context, a lower carbon footprint Cb_n results in a higher incentive η_n, and thus a higher quality score $Q_n^{\mathcal{S}}$, prioritizing clients that contribute both to model performance and to reducing the overall environmental impact.

To ensure fairness, clients that are not selected in a given round have their quality scores updated using the mechanism described in MSPCA, ensuring all clients have a chance to be chosen over time.

Algorithm 3 Low-Carbon Footprint Allocation in FL.

Require:

Refer to Shared Initial Conditions for FL Algorithms
while not converged **do**
  Follow the client allocation step as in Algorithms 1
  for each client $n \in \mathcal{S}$ **do**
    Train local model $\boldsymbol{\omega}_n$ on local dataset $\mathbb{D}_n^{\kappa_n}$
    Compute Carbon footprint Cb_n
    Assign carbon-based incentive: $\eta_n = \frac{k}{Cb_n}$
    Calculate PCA distance: $d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{center})$
    Compute quality metric: $Q_n^{\mathcal{S}} = \frac{1}{d(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{center}) \times \eta_n + \epsilon}$
  end for
  Sort clients by $Q_n^{\mathcal{S}}$ and select top $\alpha\%$ for contribution
  Update global model and client quality scores following the methods in Algorithms 1
end while

The equivalent carbon dioxide (CO_2e) metric is commonly used to calculate the carbon footprint associated with electricity generation, which varies significantly by region. As illustrated in Fig. 3, this variability can be observed across different countries such as Germany, the United Kingdom, and Sweden. Furthermore, this variability is also considered when designing and implementing carbon-efficient training for experiments.

Accurately quantifying CO_2e emissions in specific areas is challenging due to numerous influencing factors [53]–[55]. Government reports provide the following conversion rates for energy to CO_2e, highlighting the regional disparities in carbon intensity:

- In **Germany**, the carbon intensity ranges from 142.95 kg to 199.6 kg CO_2e/kWh within a single day.
- In the **United Kingdom**, it varies between 65.04 kg and 137.14 kg CO_2e/kWh per day.
- In **Sweden**, the intensity is considerably lower, ranging from 5.14 kg to 30.13 kg CO_2e/kWh per day.

Fig. 3 underscores the importance of regional approaches in assessing and managing the carbon footprints associated with power generation.

D. Randomized Double Greedy Allocation (RDGA)

The Randomized Double Greedy Allocation (RDGA) leverages submodular maximization principles to select an optimal subset of clients from a pool, aiming to balance the trade-off between model performance and carbon-conscious allocation. Given that adding more clients does not always improve the global objective due to non-monotonicity, RDGA introduces a probabilistic approach to ensure effective and eco-efficient client allocation.

Following the methodology in [44], RDGA begins by initializing two vectors: $\mathbf{a}_e = (0, 0, \ldots, 0)$, representing no clients selected, and $\mathbf{a}_f = (1, 1, \ldots, 1)$, representing all clients selected. For each client $n \in N$, the gain u_n is calculated as the difference in the utility function $U(\cdot)$ when client n is added to the subset:

$$u_n = U(\mathbf{a}_e^{-n}, a_{e,n} = 1) - U(\mathbf{a}_e).$$

Similarly, the loss v_n is computed as the reduction in the objective function when client n is removed from the full set, given by:

$$v_n = U(\mathbf{a}_f^{-n}, a_{f,n} = 0) - U(\mathbf{a}_f).$$

We specify the utility U as a weighted sum of the normalized gradients and the negative of the normalized carbon intensities:

$$U = \alpha \cdot \sum \text{Gradients} - (1 - \alpha) \cdot \sum \text{Carbon} \tag{16}$$

Here, the parameter α controls the trade-off between prioritizing model performance (e.g., gradients) and environmental sustainability (e.g., carbon intensity).

To ensure that only positive contributions are considered, RDGA computes the non-negative values:

$$u_n^+ = \max(u_n, 0) \quad \text{and} \quad v_n^+ = \max(v_n, 0)$$

The selection probability for client n is then calculated as:

$$p_n = \frac{u_n^+}{u_n^+ + v_n^+},$$

representing the likelihood of including this client in the selected subset. Based on this probability, client n is either included or excluded from the selected set:

$$a_{e,n} = 1 \text{ with } p_n \text{ or } a_{e,n} = 0 \text{ with } 1 - p_n.$$

After evaluating all clients, the final selected subset is represented by:

$$\mathbf{a}^* = \mathbf{a}_e = \mathbf{a}_f,$$

which determines the set of clients chosen for that round. The global model is subsequently updated using the local models from the clients in the selected set $\mathbf{a}^*$, and the algorithm continues iterating until convergence, as shown in Algorithm 4.

TABLE VIII: Test Results of MSPCA and APA at Various Allocation Portions ($\boldsymbol{\alpha}$) for IID ($\boldsymbol{\kappa} = \infty$) and non-IID ($\boldsymbol{\kappa} = 0.06$) Datasets.

$\boldsymbol{\alpha}$	**IID Dataset ($\boldsymbol{\kappa} = \infty$), for achieving 83% Acc.**			**non-IID Dataset ($\boldsymbol{\kappa} = 0.06$), for achieving 62% Acc.**			
Alloc. (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**	Alloc. (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**
MSPCA (Model Similarity PCA-based Client Allocation) Results							
60%	**84.36% (+1.18)**	**1.17 (-46.58%)**	**4.48 (-44.10%)**	**90%**	**65.39% (+2.71)**	2.44 (-0.20%)	7.52 (-0.32%)
90%	84.21% (+1.03)	1.68 (-23.45%)	6.31 (-21.26%)	**60%**	64.71% (+2.03)	1.36 (-42.99%)	**4.23 (-42.70%)**
50%	83.24% (+0.07)	1.26 (-42.42%)	4.81 (-39.94%)	**50%**	63.32% (+0.64)	**1.32 (-46.15%)**	5.13 (-32.00%)
Conv.	83.18%	2.19	8.01	Conv.	62.68%	2.45	7.55
80%	83.13% (-0.06)	1.98 (-9.59%)	7.55 (-5.71%)	80%	61.16% (-1.52)	NaN	NaN
RDGA	82.81%(-0.37)	NaN	NaN	70%	60.67% (-2.01)	NaN	NaN
70%	81.91% (-1.53)	NaN	NaN	RDGA	60.38% (-2.3)	NaN	NaN
APA (Accuracy-Prioritized Allocation) Results							
60%	**84.42% (+1.24)**	1.23 (-43.84%)	4.85 (-39.45%)	**60%**	**64.97% (+2.29)**	**1.62 (-33.64%)**	5.61 (-25.69%)
70%	83.22% (+0.04)	2.13 (-2.74%)	7.11 (-11.24%)	**50%**	64.70% (+2.02)	1.75 (-28.35%)	**5.27 (-30.20%)**
50%	83.19% (+0.01)	**0.58 (-73.50%)**	**2.57 (-67.91%)**	90%	63.10% (+0.42)	2.34 (-4.13%)	7.25 (-3.95%)
Conv.	83.18%	2.19	8.01	Conv.	62.68%	2.45	7.55
90%	83.17% (+0.03)	2.08 (-5.02%)	8.73 (+8.99%)	80%	62.45% (-0.23)	1.80 (-26.44%)	7.84 (+3.88%)
80%	83.08% (-0.06)	1.89 (-13.70%)	8.62 (+7.61%)	70%	60.44% (-2.24)	NaN	NaN
RDGA	82.81%(-0.37)	NaN	NaN	RDGA	60.38% (-2.3)	NaN	NaN

Algorithm 4 Randomized Double Greedy Allocation (RDGA).

Require: Refer to Shared Initial Conditions for FL Algorithms

while not converged **do**
  Initialize $\mathbf{a}_e = (0, 0, \ldots, 0), \mathbf{a}_f = (1, 1, \ldots, 1)$
  for each client $n \in N$ **do**
    Compute $u_n = U(\mathbf{a}_e^{-n}, 1) - U(\mathbf{a}_e)$, $v_n = U(\mathbf{a}_f^{-n}, 0) - U(\mathbf{a}_f)$
    Calculate $p_n = \frac{\max(u_n, 0)}{\max(u_n, 0) + \max(v_n, 0)}$
    Update $a_{e,n} = 1$ with probability p_n, else $a_{f,n} = 0$
  end for
  Final Allocation: $\mathcal{S} = \mathbf{a}^* = \mathbf{a}_e = \mathbf{a}_f$
  Train clients in $\mathcal{S}$, update global model and client scores (as in Algorithm 1)
end while
Output improved global model

E. Performance Evaluation of Three Heuristic Allocation and Randomized Double Greedy Allocation Methods

In this section, we explore the achievable optimal outputs in terms of both performance and carbon efficiency by comparing several methods: the conventional full engagement approach, three heuristic client allocation strategies, and the random double greedy algorithm. These methods are assessed using two distinct datasets: an IID dataset with no skewed distribution and another dataset with $\boldsymbol{\kappa} = 0.06$, representing the highest level of data skewness. The test results of the three heuristic allocation methods demonstrate that optimal client engagement levels enhance model accuracy and align more closely with environmental sustainability goals, as shown in Tables VIII and IX. However, the precise optimal engagement level remains undetermined. In contrast, the random double greedy allocation method exhibits significantly lower performance, primarily due to direct constraints on the performance metrics, underscoring its limitations in balancing accuracy and sustainability.

*a) **Optimal Client Allocation Engagement for Carbon-Efficient Federated Learning**:* The optimal client allocation engagement refers to the most efficient client allocation method that minimizes resource consumption to enhance carbon efficiency while maintaining high model accuracy. Striking this balance is challenging due to the need to determine the minimal level of resource allocation that satisfies accuracy requirements. This difficulty is compounded by variations in local data sizes, which affect the required computational resources, and diverse data distributions, which can lead to accuracy degradation.

Our experiments with the CINIC10 dataset demonstrate that allocating between 50% and 60% of clients achieves an effective trade-off, optimizing both carbon efficiency and model performance. This range represents the most effective allocation method, balancing sustainability with minimal impact on accuracy. However, this optimal range may vary depending on different environments, tasks, and datasets. Identifying the optimal allocation remains a significant challenge, which will be addressed by the proposed Carbon-Conscious Federated Reinforcement Learning (CCFRL) approach detailed in the next section.

*b) **Performance Analysis with IID Dataset**:* For the IID dataset ($\boldsymbol{\kappa} = \infty$), aiming to achieve 83% accuracy, the Model Similarity PCA-based Client Allocation (MSPCA) method performs best at a 60% allocation, achieving a maximum accuracy of 84.36%, with CO_2 emissions reduced by 46.58% to 1.17 kg and energy consumption reduced by 44.1% to 4.48 kWh. This allocation demonstrates a balanced approach, enhancing both accuracy and carbon efficiency. A higher 90% allocation slightly increases accuracy to 84.21%, but it also raises emissions to 1.68 kg and energy consumption to 6.31 kWh, highlighting diminishing returns in carbon efficiency compared to the 60% allocation. Notably, MSPCA at 50% allocation achieves 83.24% accuracy, with a significant re-

TABLE IX: Test Results of LCA at Various Allocation Portions ($\boldsymbol{\alpha}$) for IID ($\boldsymbol{\kappa} = \infty$) and non-IID ($\boldsymbol{\kappa} = 0.06$) Datasets.

$\boldsymbol{\alpha}$	**IID Dataset ($\boldsymbol{\kappa} = \infty$), for achieving 83% Acc.**			**non-IID Dataset ($\boldsymbol{\kappa} = 0.06$), for achieving 62% Acc.**			
Alloc. (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**	Alloc. (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**
LCA (Low carbon footprint Allocation) Results							
50%	**83.33% (+0.15)**	**0.60 (-72.60%)**	**2.20 (-72.54%)**	**60%**	**64.52% (+1.84)**	2.86 (-16.73%)	10.39 (+37.53%)
70%	83.26% (+0.08)	1.87 (-14.61%)	7.01 (-12.50%)	80%	63.74% (+1.06%)	2.35 (-4.08%)	9.61 (+27.28%)
80%	83.23% (+0.05)	1.37 (-37.44%)	6.63 (-17.20%)	90%	62.90% (+0.35%)	2.20 (-10.20%)	9.10 (+20.53%)
Conv.	83.18%	2.19	8.01	**50%**	62.78% (+0.16%)	**1.54 (-37.14%)**	**6.69 (-11.39%)**
60%	83.16% (-0.04)	1.10 (-49.77%)	6.61 (-17.46%)	Conv.	62.68%	2.45	7.55
90%	83.12% (-0.08)	2.30 (+4.93%)	8.30 (+3.62%)	70%	62.17% (-0.81%)	2.29 (-6.53%)	9.80 (+29.80%)
RDGA	82.81%(-0.37)	NaN	NaN	RDGA	60.38% (-2.3)	NaN	NaN

duction in CO_2 emissions to 1.26 kg (42.42% reduction) and energy usage to 4.81 kWh, emphasizing its potential for further carbon and energy savings with minimal impact on performance, as shown in Table VIII.

In comparison, the Accuracy-Prioritized Allocation (APA) method also performs effectively at a 60% allocation, achieving the highest accuracy of 84.42%, with CO_2 emissions of 1.23 kg (reduced by 43.84%) and energy consumption of 4.85 kWh (reduced by 39.45%). A 50% allocation under APA shows the most significant reductions in emissions and energy, achieving 83.19% accuracy with the lowest CO_2 emissions of 0.58 kg (reduced by 73.5%) and the lowest energy consumption of 2.57 kWh (reduced by 67.91%). However, this comes at the expense of only a marginal accuracy improvement, suggesting that while significant carbon and energy savings are possible, performance gains diminish at lower allocation levels, as shown in Table VIII.

Both methods indicate that a 60% allocation provides an optimal balance of high accuracy with substantial reductions in carbon emissions and energy usage, making it the most carbon-efficient strategy for IID datasets. Lower allocations, while beneficial for carbon and energy savings, highlight the trade-off between maximizing environmental impact and maintaining model accuracy, as detailed in Table VIII.

*c) **Performance Analysis with non-IID Dataset**:* For the non-IID dataset ($\boldsymbol{\kappa} = 0.06$), aiming to achieve 62% accuracy, the results indicate that the Model Similarity PCA-based Client Allocation (MSPCA) method performs best at a 90% allocation, achieving a maximum accuracy of 65.39% with a CO_2 emission of 2.44 kg and energy consumption of 7.52 kWh. However, a lower 60% allocation also demonstrates a favorable balance, yielding 64.71% accuracy, with emissions reduced by 42.99% to 1.36 kg and energy usage reduced by 42.70% to 4.23 kWh, highlighting a more carbon-efficient approach. Notably, the MSPCA method at 50% allocation achieves 63.32% accuracy with the lowest CO_2 emission of 1.32 kg, reduced by 46.15%, and moderate energy consumption of 5.13 kWh, emphasizing its potential for significantly lower carbon impact, as shown in Table VIII.

In comparison, the Accuracy-Prioritized Allocation (APA) method shows that a 60% allocation delivers the best trade-off with a maximum accuracy of 64.97%, CO_2 emissions of 1.62 kg (reduced by 33.64%), and energy consumption of 5.61 kWh, demonstrating improved accuracy and sustainability. At 50% allocation, APA achieves 64.70% accuracy, a slightly lower performance than the 60% allocation but with lower emissions of 1.75 kg and energy usage of 5.27 kWh, marking a notable reduction compared to higher allocations. Both methods emphasize the effectiveness of reduced allocation portions in achieving optimal accuracy while minimizing carbon emissions and energy consumption in non-IID environments, as shown in Table VIII.

*d) **The Low Carbon Footprint Allocation (LCA)**:* For the IID dataset ($\boldsymbol{\kappa} = \infty$) targeting 83% accuracy, the 50% allocation achieved the highest accuracy (83.33%) while significantly reducing emissions (0.60 kg CO_2, -72.60%) and energy consumption (2.20 kWh, -72.54%). Other allocations, such as 70% and 80%, also maintained high accuracy with moderate reductions in emissions and energy usage. In contrast, the conventional and 90% allocations showed increased emissions and energy consumption, with slight drops in accuracy. The Greedy approach performed the worst, achieving the lowest accuracy with missing emissions and energy data, as shown in Table IX.

For the non-IID dataset ($\boldsymbol{\kappa} = 0.06$) aiming for 62% accuracy, the 60% allocation reached the highest accuracy (64.52%) but with increased energy consumption (10.39 kWh, +37.53%) and moderate emission reductions (2.86 kg CO_2, -16.73%). The 50% allocation achieved near-target accuracy (62.78%) with the lowest emissions (1.54 kg CO_2, -37.14%) and reduced energy use (6.69 kWh, -11.39%). Conventional, 70%, 80%, and 90% allocations maintained accuracy levels close to the target with varying impacts on emissions and energy. Again, the Greedy approach yielded the lowest accuracy, with data on emissions and energy not available, as shown in Table IX.

The Randomized Double Greedy Allocation (RDGA) method is designed to optimize client selection by integrating carbon footprint constraints with performance metrics through submodular maximization, as outlined in Eq. (16). However, the direct application of these constraints significantly compromises performance, leading RDGA to underperform compared to other allocation strategies, including the conventional full allocation approach.

*e) **Randomized Double Greedy Allocation (RDGA)**:* As depicted in Tables X and IX, RDGA consistently shows lower accuracy and higher carbon emissions compared to the MSPCA, APA, and LCA methods. In the IID dataset ($\kappa = \infty$), RDGA achieves a maximum accuracy of 82.81%, notably lower than the top results from MSPCA and APA, which both exceed 84% accuracy while also achieving significant reductions in carbon emissions and energy consumption. In the non-

TABLE X: Test Results of MSPCA, APA, LCA, and RDGA for IID ($\kappa = \infty$) and non-IID ($\kappa = 0.06$) Datasets.

Method	**IID Dataset ($\kappa = \infty$), for achieving 83% Acc.**			**non-IID Dataset ($\kappa = 0.06$), for achieving 62% Acc.**			
,α %	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**	**Method**	**Max Acc.**	**Emi. (CO_2 kg)**	**Energy (kWh)**
MSPCA,60	**84.42% (+1.24)**	1.23 (-43.84%)	4.85 (-39.45%)	**APA,60**	**64.97% (+2.29)**	1.62 (-33.64%)	5.61 (-25.69%)
APA,60	84.36% (+1.16)	1.17 (-46.58%)	4.48 (-44.10%)	**MSPCA,60**	64.71% (+2.03)	**1.36 (-42.99%)**	**4.23 (-42.70%)**
LCA,50	83.33% (+0.15)	**0.60 (-72.60%)**	**2.20 (-72.54%)**	LCA,60	64.52% (+1.84)	2.86 (-16.73%)	10.39 (+37.53%)
Conv.	83.18%	2.19	8.01	Conv.	62.68%	2.45	7.55
RDGA	82.81%(-0.37)	NaN	NaN	RDGA	60.38% (-2.30)	NaN	NaN

IID dataset ($\kappa = 0.06$), RDGA's performance further declines, reaching a maximum accuracy of just 60.38%, substantially below the performance of other methods, which achieve over 64% accuracy.

These results, in line with prior findings reported in [44], underscore RDGA's limitations in balancing carbon efficiency with model performance. Although the method aims for submodular optimization, its design struggles to remain competitive under carbon constraints, resulting in a significant drop in both accuracy and sustainability metrics.

There may be potential to enhance RDGA's performance by adjusting the parameter α in Eq. (16), specifically by tuning the weights of the carbon constraints. For example, increasing the weight of carbon impact might improve balance, though this is not the primary focus of this work. In this study, we use a weight of 0.5, equally prioritizing performance and carbon impact. However, optimizing this parameter could be an avenue for future research to enhance RDGA's effectiveness.

*f) **Overall Performance Analysis**:* In distributed computing environments such as federated learning, varying levels of client engagement do not uniformly enhance performance or sustainability. Our findings show that reducing client engagement to around 50-60% optimizes both model accuracy and environmental impact by filtering out less efficient local models, improving carbon efficiency, and maintaining accuracy in the global model.

Conversely, higher engagement levels, such as 70-80%, tend to degrade performance, possibly by excluding high-quality models that significantly contribute to both accuracy and carbon efficiency. Interestingly, a 90% engagement ratio consistently outperforms full client engagement in terms of accuracy but provides minimal improvements in carbon efficiency. These results highlight the importance of carefully balancing client participation, as improper engagement levels can lead to reduced model accuracy and increased environmental impact.

The optimal engagement range of 50-60% has proven effective in our testing environment, which includes scalable datasets like CINIC-10, a flexible number of clients (e.g., 50), and neural network architectures such as DenseNet-121 and ResNet-18. However, this optimal level may vary in different contexts. Factors such as dataset characteristics, model diversity, and carbon footprint must be considered when determining the best client engagement strategy in federated learning systems. These findings emphasize the complexity of selecting the optimal engagement level without a thorough understanding of the specific environment.

Table X illustrates the varying effectiveness of different allocation strategies across IID ($\kappa = \infty$) and non-IID ($\kappa = 0.06$) datasets. The Model Similarity PCA-based Client Allocation (MSPCA) method achieves the highest accuracy for the IID dataset, with a 1.24% improvement over the baseline, alongside significant reductions in emissions (43.84%) and energy consumption (39.45%). However, this strategy is not always the best option in other contexts. For the non-IID dataset, the Accuracy-Prioritized Allocation (APA) method achieves the highest accuracy (64.97%) while also reducing emissions (33.64%) and energy consumption (25.69%). Meanwhile, the Low-Carbon Footprint Allocation (LCA) method significantly minimizes emissions and energy use, particularly in IID settings, though it sacrifices some performance in non-IID environments. This underscores that no single allocation strategy consistently excels across all scenarios.

To address these challenges, we present the Carbon-Conscious Federated Reinforcement Learning (CCFRL) framework, which incorporates an adaptive client allocation scheme that dynamically adjusts client engagement based on real-time performance and carbon efficiency metrics. CCFRL continuously adapts to changes in dataset variability, performance metrics, and environmental impact, optimizing both model accuracy and carbon footprint. By integrating carbon-aware state representations and adaptive client allocation into the training loop, CCFRL effectively balances performance with sustainability, ensuring an efficient federated learning process that aligns client participation with both accuracy and environmental objectives.

V. Carbon-Conscious Federated Reinforcement Learning (CCFRL)

This research introduces the Carbon-Conscious Federated Reinforcement Learning (CCFRL) framework, which integrates carbon footprint data into state representation to optimize both performance and carbon efficiency in federated learning environments. The state representation is dynamic, incorporating key metrics such as accuracy, model similarity, and carbon emissions, allowing it to adapt to changing conditions.

Section IV-E demonstrates that while accuracy and model similarity effectively identify high-performing clients, determining the optimal client participation in training often demands manual adjustments. To address this, we propose two novel methods: Accuracy Priority Allocation (APA) integrated with carbon emissions and Model Similarity PCA-based Client Allocation (MSPCA) with carbon emissions. These methods illustrate the intricate balance between performance and

carbon efficiency, highlighting the need for adaptive state representation to manage these trade-offs.

Balancing performance with carbon efficiency poses challenges, as enhancing performance can lead to higher carbon emissions. The CCFRL framework addresses this by employing a flexible state representation that adjusts dynamically to balance these competing objectives. Transition mechanisms guide state changes to optimize both carbon efficiency and performance, utilizing an innovative reward system that adapts to fluctuations in the environment, such as changes in carbon intensity and performance metrics.

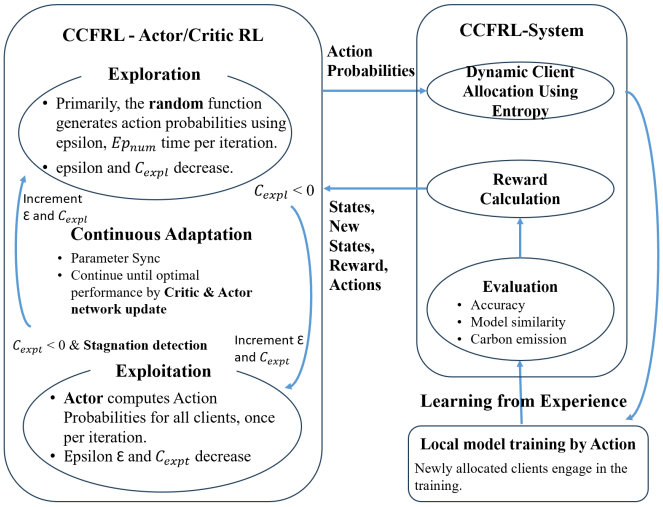


Fig. 4: CCFRL System Architecture: Actor/Critic RL with Dynamic Client Allocation and Stagnation Detection.

To effectively manage the dynamic environment characterized by fluctuating carbon intensity and performance variations, the CCFRL framework employs a balanced exploration-exploitation strategy, as depicted in Fig. 4. This approach dynamically adjusts client engagement ratios through continuous adaptation, allowing the system to respond to changes and maintain optimal performance while minimizing carbon emissions. By leveraging exploration during uncertain conditions and exploitation when optimal parameters are detected, CCFRL ensures a strategic balance that adapts to evolving states.

The framework's adaptive mechanisms, including entropy-based client allocation and stagnation detection, enable it to navigate complex trade-offs between performance and carbon efficiency. As a result, CCFRL selectively engages clients that align with the global model's objectives, optimizing resource use and significantly reducing the overall carbon footprint without compromising performance; additionally, notations used are summarized in Table XI.

A. Sustainable Model Coherence Policy (SMCP)

The Sustainable Model Coherence Policy (SMCP) provides a strategic framework for directing the behavior of a Reinforcement Learning (RL) agent in a federated learning environment. This policy aims to enhance model coherence across distributed clients while simultaneously minimizing their carbon emissions. By integrating model similarity metrics and environmental considerations, this policy supports the dual objectives of maintaining high model fidelity and promoting sustainability.

TABLE XI: Notation Summary in CCFRL.

Notation	Description
S_t	State at timestep t
$\boldsymbol{\omega}_{n_t}$	Local model parameters of client n_t
$\boldsymbol{\omega}_{center}$	Centroid of PCA-transformed model weights
$d(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}_{center})$	PCA-based distance bet. local and central model
Cb_{n_t}	Carbon emissions of client n_t
$\text{Acc}(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega})$	Accuracy of local model n_t vs. global model
R_t	Reward at timestep t
$\beta_{weighting}$	Weight bet. model coherence and carbon emissions
A_t	Action taken by the RL agent at t
H_{norm}	Normalized entropy of client allocation
S_{alloc}	Number of allocated clients
a, b	Coefficients for entropy-based allocation
Ep_{num}	Exploration iterations
Ct_{expl}	Exploration counter
Ct_{expt}	Exploitation counter
N_{thres}	Stagnation detection threshold
t	Timestep
γ	Discount factor

In this policy, each state captures crucial information that guides the RL agent's decisions at any given timestep (t):

*a) **State Representation** S_t:* : The state S_t at each timestep is structured as:

$$S_t = \left(\left[d(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}_{center}) \right], \left[Cb_{n_t} \right] \right),$$

where $d(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}_{center})$ represents the distance derived from a PCA-based transformation, quantifying the dissimilarity between the local model parameters $\boldsymbol{\omega}_{n_t}$ of client n_t and the centroid $\boldsymbol{\omega}_{center}$ of the PCA-transformed model weights, which are computed using the contributions from all participating clients during training. This measure helps assess the alignment or coherence of the local model with the central model. Cb_{n_t} measures the carbon emissions produced by the computational activities of the client n_t, reflecting the environmental impact of training. Alternatively, Cb_{n_t} could be the carbon intensity measured in the location, to which a client belongs.

This dual-state configuration allows the RL agent to consider both the coherence of models across different clients and the sustainability of their operations, facilitating decisions that optimize both technological and environmental outcomes.

*b) **Action and Reward Mechanism**:* The RL agent's actions, based on the current state, strive to harmonize model coherence and environmental sustainability. Each action involves selecting clients whose data contributions are anticipated to optimize both the global model's consistency and minimize carbon emissions. Upon action execution, these selected clients engage in the model training phase, updating the global model parameters $\boldsymbol{\omega}$, and thereby directly impacting both the model's accuracy and its ecological footprint.

The reward function, which plays a pivotal role in guiding

these decisions, is formulated as follows:

$$R_t = \beta_{\text{weighting}} \left(\frac{d(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}_{\text{center}_t}) - d(\boldsymbol{\omega}_{n_{t+1}}, \boldsymbol{\omega}_{\text{center}_{t+1}})}{d(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}_{\text{center}_t})} \right) + (1 - \beta_{\text{weighting}}) \left(\frac{Cb_{n_t} - Cb_{n_{t+1}}}{Cb_{n_t}} \right), \tag{17}$$

where $\beta_{weighting}$ adjusts the priority between reducing the PCA distance (thus enhancing model coherence) and decreasing carbon emissions. It calculates the relative decrease from the previous state, making the reward sensitive to the scale of changes and encouraging proportionate improvements.

B. Eco-Accuracy Optimization Policy (EAOP)

The Eco-Accuracy Optimization Policy provides a structured framework for governing the behavior of a Reinforcement Learning (RL) agent within a federated learning environment. This policy is designed to balance two critical objectives: prioritizing accuracy in model training and minimizing the carbon footprint associated with the computational processes. By integrating environmental considerations directly into the decision-making process, this policy supports sustainable development goals alongside technological advancement.

In this RL framework, the state representation is a critical component that encapsulates key aspects of the learning environment at any given timestep (t):

*a) **State Representation** S_t:* : Each state S_t is represented as a tuple:

$$S_t = \left([\text{Acc}(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega})], [Cb_{n_t}] \right),$$

where $\text{Acc}(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega})$ measures the accuracy of the local model $\boldsymbol{\omega}_{n_t}$ relative to the global model $\boldsymbol{\omega}$, providing a direct measure of performance. Cb_{n_t} quantifies the carbon emissions generated by client n_t during computation, reflecting the environmental impact.

This dual-state system allows the RL agent to assess both the performance of local models and their associated carbon emissions, facilitating informed and balanced decision-making.

*b) **Action and Reward Mechanism**:* The RL agent's actions are meticulously computed at each timestep t by the actor-network, which assesses the current state and outputs a set A_t of action probabilities for each client. These probabilities guide the allocation of clients, aiming to optimally balance improvements in model accuracy with reductions in carbon emissions.

Following the determination of actions, the selected clients participate in the model training process. This participation updates the global model parameters $\boldsymbol{\omega}$, affecting both model accuracy and carbon emissions. The new state S_{t+1} is then characterized by these updated parameters, reflecting changes in performance metrics such as accuracy and emissions resulting from the training conducted by the allocated clients.

The reward function is crucial for guiding these decisions and is defined as follows:

$$R_t = \beta_{weighting} \left(\text{Acc}(\boldsymbol{\omega}_{n_{t+1}}, \boldsymbol{\omega}) - \text{Acc}(\boldsymbol{\omega}_{n_t}, \boldsymbol{\omega}) \right) + (1 - \beta_{weighting}) \left(\frac{1}{Cb_{n_{t+1}}} - \frac{1}{Cb_{n_t}} \right),$$

where $\beta_{weighting}$ is a weight balancing the significance of accuracy improvement (Acc) against the reduction of carbon emissions (C). This reward formula incentivizes the agent to select actions that not only bolster the model's performance but also reduce its environmental impact.

Through this dynamic interplay between action determination, client-based model training, and state transitions, the Eco-Accuracy Optimization Policy directs the RL agent towards strategies that ensure training aligns with both performance enhancement and sustainability goals. By integrating these elements, the policy fosters a responsible approach to deploying machine learning across distributed networks, optimizing resource utilization and minimizing ecological footprints.

C. Stagnation Detection Scheme in Reinforcement Learning Exploration

The Carbon-Conscious Federated Reinforcement Learning (CCFRL) includes a stagnation detection scheme to address performance stagnation during RL exploitation phases. This scheme enhances the RL agent's adaptability by dynamically balancing exploration and exploitation based on real-time performance assessments.

1) Exploration Phase: In the exploration phase, the RL agent engages in multi-exploration mode, testing various client configurations to find optimal model updates. This phase lasts for a set count, $N_{explore}$ (e.g., 50), after which the agent switches to exploitation, focusing on high-performing configurations.

2) Exploitation Phase and Stagnation Detection: During exploitation, the RL agent uses previously identified configurations to refine the model. To detect stagnation, the agent applies a t-test, comparing recent and past accuracy data to assess whether performance has plateaued.

*a) **Stagnation Detection Process**:* The t-test calculates the t-statistic:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the means, s_1^2 and s_2^2 are the variances, and n_1 and n_2 are the sample sizes of recent and past data.

*b) **Evaluation Criteria**:*

- **p-value** $> N_{thres}$ **(e.g., 0.05)**: A p-value greater than the threshold N_{thres} indicates no statistically significant difference between recent and past performance, suggesting that the model's performance has stagnated. In this scenario, the RL agent remains in exploitation mode, focusing on the current configurations, until the exploitation counter (Ct_{expt}) depletes to zero. If no improvement is detected during this period, the agent reverts to the exploration phase by increasing the exploration rate

(epsilon) significantly, prompting the agent to search for new or better-performing client configurations.

- **p-value** $\leq N_{thres}$: A p-value less than or equal to the threshold N_{thres} indicates a statistically significant change in performance, which could represent either improvement or degradation. If degradation is detected, the agent promptly returns to the exploration phase to prevent further performance decline and to identify alternative configurations that could enhance model accuracy and carbon efficiency.

*c) **Dynamic Memory Management**:* Upon detecting stagnation or performance degradation, the RL agent dynamically manages its memory by selectively pruning outdated learned experiences. This process removes irrelevant or less useful data, ensuring that the agent does not rely on obsolete performance trends that no longer reflect the current training environment, thus enhancing decision-making and responsiveness.

3) Transition Mechanisms: Transitions between exploration and exploitation involve adjusting parameters like the exploration rate (epsilon), promoting adaptability in response to performance stagnation and degradation.

4) Significance: The stagnation detection scheme allows the RL agent to adjust strategies dynamically, balancing exploration and exploitation effectively. This approach prevents prolonged suboptimal performance, enhancing both model accuracy and carbon efficiency in federated learning environments.

D. Entropy-Based Dynamic Client Allocation

Dynamically allocating clients based on the entropy of their action probabilities enhances learning efficiency across distributed environments. Entropy is a measure of randomness or uniformity in the assignment probabilities of each client. It is mathematically defined as $H = -\sum_{n=1}^{N} p_n \log(p_n)$, where p_n represents the probability of allocating the n-th client, and N is the total number of clients. To facilitate comparison across different client counts and probability distributions, entropy is normalized using $H_{\text{norm}} = \frac{H}{\log(N)}$.

In this approach, the allocation of clients is adaptively adjusted based on the level of entropy: higher entropy, indicating similar preferences among clients, leads to fewer allocations, assuming that a smaller set of clients can effectively represent the whole. Conversely, lower entropy, indicating a more varied distribution of client preferences, results in the allocation of more clients to capture a wider range of data and perspectives. This dual method is encapsulated in the allocation threshold S, which is formulated as:

$$S_{alloc} = N \cdot (a + b \cdot (1 - H_{\text{norm}})) \tag{18}$$

Here, a and b are coefficients where a sets the base rate of allocation and b adjusts how the allocation sensitivity inversely correlates with entropy. This method ensures a balanced allocation policy that promotes exploration by incorporating diverse client data when entropy is low and exploitation by focusing on a representative subset when entropy is high. This adaptive allocation optimizes resource use and maximizes learning outcomes in federated environments.

E. Learning Objective and Algorithm

The goal of this research is to maximize the effectiveness and sustainability of federated learning deployments through a sophisticated Reinforcement Learning (RL) framework. The framework employs two distinct policies — the Eco-Accuracy Optimization Policy and the Sustainable Model Coherence Policy — each designed to address dual-objective optimization challenges. The learning objective and the algorithmic approach for each policy are crafted to ensure that both performance and environmental metrics are optimized throughout the learning process.

*a) **Learning Objective**:* For both methods, the fundamental learning objective is formulated as:

$$\max \sum_{t=0}^{T} \gamma^t R_t,$$

where γ is the discount factor, which determines the weighting of immediate versus future rewards. This objective encourages the RL agent to optimize actions over the course of multiple interactions, enhancing the model's performance and reducing its environmental impact over time. The aim is to develop a policy that effectively balances accuracy, model coherence, and carbon emissions, adapting dynamically to changes in client data distributions and computational capacities.

Algorithm 5 Eco-Optimization via RL with Entropy-Based Dynamic Client Allocation and Stagnation Detection.

Require: Global model $\boldsymbol{\omega}$, $\gamma = 0.95$, exploration counter Ct_{expl}, exploitation counter Ct_{expt}, exploration number Ep_{num}, stagnation threshold N_{thres}.

Initialize actor, critic networks, and replay memory $\mathcal{D}$
Initialize state S_0 using data metrics (e.g., accuracy, PCA, carbon emissions)
Set explore_mode ← True
while not converged **do**
  if explore_mode **then**
    $A_t \leftarrow$ random policy; repeat for Ep_{num} times per iteration
  else
    $A_t \leftarrow$ policy decision from $\pi(S_t)$
  end if
  Calculate entropy H_{norm} and allocation factor $\boldsymbol{\alpha}$
  Determine $S_{alloc} \leftarrow \lceil \boldsymbol{\alpha} \times N \rceil$, execute on top S_{alloc}
  Execute A_t, observe R_t, S_{t+1}, store (S_t, A_t, R_t, S_{t+1}) in $\mathcal{D}$
  Decrement corresponding counter (Ct_{expl} or Ct_{expt}), adjust ϵ
  if counter reaches zero or stagnation_detection() **then**
    Toggle explore_mode, reset counters, adjust ϵ
  end if
end while
return $\boldsymbol{\omega}$

function STAGNATION_DETECTION
  Compare recent and past performance with t-test
  return True if p-value $> N_{thres}$, else False
end function

*b) **Algorithm Description**:* The RL framework follows a structured approach to ensure robust and adaptive learning within an Actor-Critic architecture, incorporating entropy-based dynamic client allocation and stagnation detection:

- Initialization:
  - Set initial global model parameters $\boldsymbol{\omega}$ and discount factor γ.
  - Initialize actor and critic networks with random weights to facilitate unbiased exploration.
  - Establish the replay memory $\mathcal{D}$ with capacity M to store learning transitions.
  - Initialize the state S_0 using initial data metrics such as accuracies, PCA distances, and carbon emissions.
  - Set exploration mode with initial counts Ct_{expl} and exploitation counters Ct_{expt}.
- Exploration and Exploitation Phases:
  - During the exploration phase, the RL agent explores various client configurations until Ct_{expl} reaches zero, prompting a switch to exploitation mode.
  - In exploitation mode, the RL agent refines the model using identified configurations and dynamically checks for performance stagnation and degradation through statistical analysis.
- Dynamic Client Allocation Using Entropy:
  - Entropy Calculation: Calculate normalized entropy $H_{\text{norm}} = \frac{-\sum_{n=1}^{N} p_{n,t} \log(p_{n,t})}{\log(N)}$ to assess the distribution of client selection probabilities.
  - Client Allocation: Compute allocation factor $\boldsymbol{\alpha} = 1 - H_{\text{norm}}$ and determine the number of clients $S_{alloc} = \lceil \boldsymbol{\alpha} \times N \rceil$ for action execution based on their probabilities.
- Stagnation and Performance Degradation Detection:
  - Continuously monitor performance using a t-test comparing recent and past accuracy records.
  - p-value $> N_{thres}$: Indicates stagnation; the agent remains in exploitation mode until the counter Ct_{expt} reaches zero. If no improvement occurs, the agent switches to exploration with a higher exploration rate ϵ.
  - p-value $\leq N_{thres}$: Indicates a significant performance change, either improvement or degradation. If degradation is detected, the agent promptly switches back to exploration to identify better configurations.
- Reward Calculation and Policy Update:
  - Execute the chosen action A_t, observe the resulting reward R_t, and update the state to S_{t+1}.
  - Store the transition (S_t, A_t, R_t, S_{t+1}) in replay memory $\mathcal{D}$.
  - Sample minibatches from $\mathcal{D}$ to update the actor and critic networks.
- Learning from Experience:
  - Update the critic by minimizing the loss: $Loss = \frac{1}{M} \sum (y_j - V(S_j; \theta))^2$, where $y_j = R_j + \gamma V(S_{j+1}; \theta)$.
  - Update the actor policy using the policy gradient to refine future actions based on the critic's evaluations.
- Continuous Adaptation:
  - Synchronize the target critic parameters with the critic network parameters periodically every C steps.
  - Repeat the process until convergence or a predefined number of learning cycles, resulting in the optimized global model $\boldsymbol{\omega}$.

By integrating these algorithmic steps, the RL framework not only targets improvements in federated learning efficacy and efficiency but also ensures that the learning process is aligned with sustainability goals. The combined approach of using specific strategies for accuracy and coherence, alongside a consistent focus on reducing carbon emissions, enables a comprehensive and adaptive optimization of federated learning systems (see Algorithm 5).

VI. Performance analysis of Carbon-Conscious Federated Reinforcement Learning in Dynamic Environments

This section presents a performance analysis of Carbon-Conscious Federated Reinforcement Learning (CCFRL), comparing it with other approaches, including a conventional method involving full training engagement and three heuristic client allocation methods. The comparison focuses on the dynamic adaptation capabilities of these methods in challenging environments, considering factors like carbon intensity variations and performance degradation, as observed in the time plots.

A. Experimental setting

To enable the reproduction of the test results; the test configuration is as follows:

- The experiments were conducted using NVIDIA A100 GPUs, hosted on the Berzelius supercomputing resource at the National Supercomputer Centre. Each model was trained for a minimum of three days, totaling over 18,000 GPU hours.
- Energy Consumption Calculation: The total energy consumption was calculated using the formula $E_{\text{total}} = \sum_{n=1}^{N} P_{\text{avg},n} \cdot T_{\text{run},n}$, where $P_{\text{avg},n}$ represents the average power usage per NVIDIA A100 GPU, measured at 300Wh during the experiment.
- Carbon Emission: Carbon intensity data from January 1-4, 2023, was analyzed for Germany (142.95-479.6 g CO_2e/kWh), the United Kingdom (41.84-179.45 g CO_2e/kWh), and Sweden (1.89-30.13 g CO_2e/kWh).
- Neural Network Architecture and Dataset: We employed the DenseNet-121 and ResNet-18 architectures and the CINIC-10 dataset, which includes 10 labels [50].
- Datasets: Four datasets with different data distributions were created using varying $\boldsymbol{\kappa}$ values to control non-IID characteristics through Kullback-Leibler (KL) divergence: 0.06 (mean: 13.187, SD: 3.406), 0.1 (mean: 10.768, SD: 3.286), 0.2 (mean: 6.583, SD: 2.779), and 0.5 (mean: 1.911, SD: 1.356). Additionally, a fully IID dataset was generated with $\boldsymbol{\kappa} = 100$ (mean: 0.00349, SD: 0.001621).

TABLE XII: Integrated Performance Analysis of SMCP, EAOP, Conventional full engagement (Conv.), and MSPCA with various distributions (i.e., 50% to 90%) on Datasets with $\boldsymbol{\kappa} = \infty, 0.5, 0.1, 0.06$ for achieving 83%, 80%, 71%, and 62%.

MSPCA	**IID Dataset $\boldsymbol{\kappa} = \infty$, for achieving 83% Accuracy**			**MSPCA**	**non-Dataset $\boldsymbol{\kappa} = 0.5$, for achieving 80% Accuracy**		
$\boldsymbol{\alpha}$ (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**	**$\boldsymbol{\alpha}$ (%)**	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**
60%	**84.36% (+1.18)**	**1.17 (-46.67%)**	**4.48 (-44.10%)**	**EAOP**	**82.16% (+0.59)**	**1.48 (-9.76%)**	5.56 (-13.28%)
EAOP	84.34% (+1.16)	**1.66 (-24.11%)**	6.17 (-22.99%)	**SMCP**	82.15% (+0.58)	**1.46 (10.97%)**	5.48 (-14.53%)
SMCP	84.27% (+1.09)	**1.25 (-42.80%)**	5.83 (-27.21%)	90%	82.10% (+0.53)	1.47 (-10.37%)	5.39 (-15.93%)
90%	84.21% (+1.03)	1.68 (-23.46%)	6.31 (-21.20%)	**60%**	82.09% (+0.52)	**1.15 (-29.88%)**	**4.46 (-30.42%)**
50%	83.24% (+0.07)	1.26 (-42.37%)	4.81 (-39.96%)	Conv.	81.57%	1.64	6.41
Conv.	83.18%	2.19	8.01	80%	80.95% (-0.62)	1.63 (-0.61%)	6.26 (-2.34%)
80%	83.13% (-0.06)	1.98 (-9.54%)	7.55 (-5.73%)	50%	80.94% (-0.63)	1.14 (-30.49%)	4.45 (-30.58%)
70%	81.91% (-1.53)	NaN	NaN	70%	79.98% (-1.59)	NaN	NaN
MSPCA	**non-Dataset $\boldsymbol{\kappa} = 0.1$, for achieving 71% Accuracy**			**MSPCA**	**non-Dataset $\boldsymbol{\kappa} = 0.06$, for achieving 62% Accuracy**		
$\boldsymbol{\alpha}$ (%)	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**	**$\boldsymbol{\alpha}$ (%)**	**Max Acc.**	**Emission (CO_2 kg)**	**Energy (kWh)**
SMCP	**72.44% (+1.32)**	**1.86 (-61.78%)**	**6.41 (-64.23%)**	**90%**	**65.39%** (+2.71)	2.44 (0.00%)	7.52 (-0.27%)
EAOP	71.91% (+1.11)	**2.00 (-58.81%)**	8.41 (-53.07%)	**EAOP**	65.25% (+2.02)	**2.35 (-3.69%)**	6.89 (-8.62%)
60%	71.62% (+0.70)	2.25 (-53.74%)	8.22 (-54.12%)	**SMCP**	64.55% (+1.25)	**1.81 (-25.82%)**	5.12 (-32.10%)
Conv.	71.12%	4.85	17.91	**60%**	64.70% (+1.56)	**1.36 (-44.26%)**	**4.23 (-43.92%)**
90%	71.04% (-0.11)	4.40 (-9.35%)	16.11 (-10.04%)	50%	63.32% (+0.64)	1.32 (-46.02%)	5.12 (-32.10%)
80%	70.62% (-0.50)	NaN	NaN	Conv.	62.68%	2.44	7.54
50%	69.94% (-1.65)	NaN	NaN	70%	61.16% (-1.52)	NaN	NaN
70%	69.56% (-2.19)	NaN	NaN	80%	60.67% (-2.01)	NaN	NaN

- Normalization of State Values S_t: State values were normalized to align with model outputs. The local model accuracy, ranging from 0 to 1, and the PCA-based model similarity, from 0.1 to 4, were divided by their maximums, capped at 4, for uniformity across datasets.
- Dynamic Allocation: Dynamic allocation was guided by entropy settings with parameters $a = 0.5$ and $b = 0.2$ in Eq. 18, adjusting the allocation between 50% and 100%. Reward efficiency utilized $\beta_{weighting}$ values of {0.5, 0.6, 0.7, 0.8} as specified in Eq. 17.
- Stagnation Detection Setting: A stagnation threshold $N_{\text{thres}} = 0.05$ is used to identify instances of stagnation and performance degradation, as described in V-C2b.
- Threshold to Prevent Frequent Fluctuations: To minimize frequent switching between exploration and exploitation, the minimum number of steps is set to $Ct_{\text{expl}} = 10$ for exploration and $Ct_{\text{expt}} = 200$ for exploitation, as specified in Algorithm 5.

B. Performance Analysis of Dynamic Training Engagement in Client Allocation on DenseNet-121

Table XII compares maximum accuracy, carbon emissions, and energy consumption across various data distributions using SMCP, EAOP, MSPCA, and full engagement. SMCP and EAOP excel in reducing emissions while maintaining high performance through adaptive client allocation.

Dynamic client allocation is facilitated by adaptive exploration and exploitation transitions, incorporating t-tests and entropy-based adjustments from RL actions to determine the optimal number of participating clients. This approach allows the system to learn and respond to environmental changes, overcoming performance challenges. Entropy, which measures randomness in model training, plays a key role in optimizing client participation, enhancing both model quality and sustainability. This adaptive process balances performance with carbon efficiency by minimizing emissions while maintaining robust model outcomes.

The analysis demonstrates the effectiveness of the proposed methods, leveraging advanced state representations, reward mechanisms, dynamic exploration-exploitation transitions, and entropy-based client engagement, all integrated through reinforcement learning. These strategies significantly reduce carbon footprints without compromising model performance. Unlike heuristic client allocation strategies that depend on fixed portions (e.g., 50%-90%), the integrated approach dynamically determines the optimal client allocation, eliminating the need for manual adjustments.

1) Performance Analysis of Client Allocation in Balanced and Low-skewed Distributions $\boldsymbol{\kappa} = \{\infty, 0.5\}$: Table XII provides a detailed comparison of the performance metrics—maximum achievable accuracy, carbon emission footprint (CO_2 kg), and energy consumption (kWh)—for datasets with balanced ($\boldsymbol{\kappa} = \infty$) and low-skewed ($\boldsymbol{\kappa} = 0.5$) distributions.

Both SMCP and EAOP achieve superior accuracy compared to other methods, with SMCP reaching up to 84.27% and EAOP achieving 84.34% on balanced data ($\boldsymbol{\kappa} = \infty$). In the low-skewed distribution ($\boldsymbol{\kappa} = 0.5$), EAOP and SMCP maintain high accuracy (82.16% and 82.15%, respectively), demonstrating their robustness across different data distributions. SMCP and EAOP notably reduce carbon emissions compared to conventional full engagement methods. In balanced datasets, SMCP reduces emissions by up to 42.80%, while EAOP achieves a 24.11% reduction. For low-skewed data, EAOP and SMCP also lower emissions, with EAOP achieving a reduction of 9.76%, highlighting the environmental benefits of these policies. Energy consumption is significantly reduced with dynamic methods. On balanced data, SMCP achieves a 27.21% reduction in energy use, and EAOP reduces it by 22.99%. In low-skewed data, these methods continue to

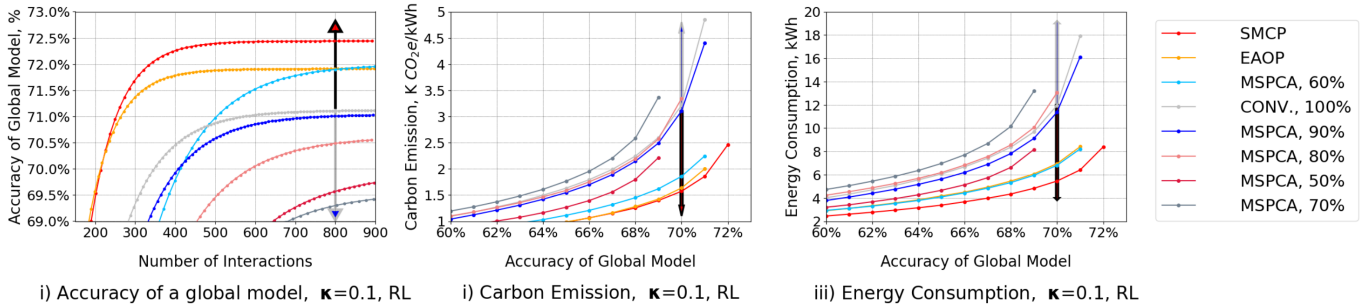


Fig. 5: Performance evaluation of SMCP, EAOP, MSPCA, and Conventional Full Engagement (Conv.) on a highly skewed dataset ($\kappa = 0.1$) shows that SMCP delivers the highest accuracy at 72.44%, while reducing emissions by 61.78% and energy consumption by 64.23% compared to the conventional method. EAOP achieves 71.91% accuracy, with a 58.81% reduction in emissions and a 53.07% decrease in energy consumption.

offer energy savings, with EAOP achieving a reduction of 13.28% and SMCP 14.53%, indicating that dynamic client allocation not only enhances performance but also improves energy efficiency.

2) Performance Analysis of Client Allocation in High-Skewed Distributions $\kappa = \{0.1, 0.06\}$: Table XII provides a comprehensive evaluation of performance metrics—maximum accuracy, carbon emissions (CO_2 kg), and energy consumption (kWh)—for high-skewed datasets with $\kappa = 0.1$ and $\kappa = 0.06$. Figure 5 specifically illustrates the results for $\kappa = 0.1$, showing that SMCP achieves the highest accuracy at 72.44%, with a 61.78% reduction in carbon emissions and a 64.23% decrease in energy consumption compared to the conventional method. EAOP closely follows with 71.91% accuracy, reducing emissions by 58.81% and energy usage by 53.07%.

For $\kappa = 0.06$, SMCP and EAOP maintain strong performance, achieving accuracies of 65.39% and 65.25%, respectively. SMCP reduces emissions by 25.82%, and EAOP achieves an 8.62% reduction in energy consumption, showcasing their adaptability in handling extreme data distributions. These results demonstrate the effectiveness of dynamic client allocation strategies, particularly SMCP and EAOP, in enhancing model performance and sustainability.

Both SMCP and EAOP significantly outperform full client engagement methods by dynamically adjusting client participation using reinforcement learning and adaptive exploration-exploitation transitions. This approach allows them to optimize performance while minimizing carbon emissions and energy use without manual intervention. Unlike fixed allocation strategies, these methods adapt in real time to varying environmental conditions and data characteristics, maximizing both accuracy and sustainability across different skewness levels ($\kappa = \{\infty, 0.5, 0.1, 0.06\}$).

C. Performance Analysis of Dynamic Training Engagement on ResNet-18

This section analyzes the dynamic training engagement strategies on ResNet-18, focusing on the Sustainable Model Coherence Policy (SMCP), Model Similarity PCA-based Client Allocation (MSPCA), and conventional full engagement (Conv.) under highly skewed data distributions ($\kappa = 0.1$ and $\kappa = 0.06$).

Table XIII compares SMCP and other client allocation methods across varying participation levels for $\kappa = 0.1$ and $\kappa = 0.06$. For $\kappa = 0.06$, SMCP achieves a maximum accuracy of 66.97%, surpassing the conventional approach by 1.02% while reducing carbon emissions by 24.16% and energy consumption by 9.53%. These results highlight SMCP's ability to enhance accuracy and significantly lower environmental impact compared to conventional methods. Notably, a 50% client participation rate yields significant reductions in carbon emissions (66.1%) and energy consumption (58.01%), illustrating that considerable sustainability improvements can be achieved even with reduced client engagement.

TABLE XIII: Comparative Performance Analysis of SMCP, MSPCA, and the Conventional Method on ResNet-18 for Datasets with Skewness Levels $\kappa = 0.06$ and $\kappa = 0.1$, Evaluated at Target Accuracies of 64% and 70%, Respectively.

Methods	**Maximum**	**$\kappa = 0.06$, for achieving 64% Acc.**	
MSPCA	**Accuracy**	**Emission (CO_2 kg)**	**Energy (kWh)**
SMCP	**66.97%** (+1.02)	**2.04 (-24.16%)**	7.88 (-9.53%)
Conv.	65.95%	2.69	8.71
70%	65.73% (-0.22)	1.30 (-51.67%)	4.95 (-43.18%)
50%	65.59% (-0.36)	**0.91 (-66.17%)**	**3.66 (-58.01%)**
80%	65.40% (-0.55)	2.34 (-13.02%)	7.43 (-14.72%)
60%	64.99% (-0.96)	1.44 (-46.47%)	5.11 (-41.34%)
90%	63.90% (-2.05)	NaN	NaN
Methods	**Maximum**	**$\kappa = 0.1$, For achieving 70% Acc.**	
MSPCA	**Accuracy**	**Emission (CO_2 kg)**	**Energy (kWh)**
SMCP	**71.46% (+0.80)**	**1.39 (-1.41%)**	5.25 (-3.84%)
60%	71.04% (+0.38)	1.38 (-2.13%)	5.09 (-6.78%)
70%	70.76% (+0.10)	1.37 (-2.84%)	5.66 (+3.66%)
Conv.	70.66%	1.41	5.46
50%	70.23% (-0.43)	**1.33 (-5.67%)**	**4.85 (-11.19%)**
90%	70.14% (-0.52)	2.51 (+78.01%)	8.50 (+55.68%)
80%	69.95% (-0.71)	NaN	NaN

In the case of $\kappa = 0.1$, SMCP achieves a maximum accuracy of 71.46%, with slight improvements over other methods in terms of accuracy (+0.80%). Notably, the SMCP method shows consistent performance in reducing carbon emissions and energy consumption, achieving reductions of 1.41% and

3.84%, respectively. However, as participation levels decrease to 50%, the reductions in energy consumption reach up to 11.19%, highlighting the adaptability of SMCP in managing trade-offs between accuracy and environmental impact.

D. Discussion

Future enhancements of the CCFRL framework will focus on several key areas to further improve its adaptability and performance:

1) Enhancing State Representations and Transition Functions: CCFRL's dynamic state representation adapts to changes in client performance and environmental conditions. However, optimizing decision-making requires deeper insights into how performance and carbon efficiency metrics affect transitions within the Markov decision process (MDP). Future research should refine the state representation to capture subtle interactions, such as the evolving trade-off between carbon emissions and accuracy, for better optimization across various conditions.

2) Introducing Additional State Features: The current state representation in CCFRL is robust but could be enhanced by integrating features like energy consumption, client reliability, and network latency. These additions would improve decision-making in complex federated learning environments and help maintain strong performance as CCFRL scales to diverse datasets and conditions.

3) Refining Reward Mechanisms: A core challenge of CCFRL is defining a reward mechanism that balances performance and carbon efficiency. Due to varying conditions in federated learning environments, reward normalization can be difficult, leading to suboptimal outcomes. Future work should explore adaptive reward mechanisms, possibly using meta-learning, to dynamically adjust rewards based on the current state. This would improve reward tuning and ensure the system consistently prioritizes both objectives.

4) Enhancing Entropy-Based Client Allocation: The entropy-based client allocation in CCFRL effectively balances exploration and exploitation, but the assumption that low entropy always corresponds to a 60% client engagement rate may not suit all datasets and environments. A more granular approach, incorporating dataset-specific adjustments and additional state features, is needed to better tailor client participation rates.

5) Balancing Exploration and Exploitation: The balance between exploration and exploitation in CCFRL is crucial for adaptability in dynamic environments. Further refinement is needed to optimize transitions between these modes. In fluctuating carbon intensity environments, a more flexible exploration strategy that considers both short-term and long-term impacts of client engagement could enhance system performance. Dynamic adjustments based on real-time performance and carbon data would further improve efficiency.

6) Geographic Generalizability: The current experimental setup relies on carbon intensity data from only a few countries (Germany, UK, and Sweden), limiting the generalizability of the results. Expanding the geographic scope by using platforms like Electricity Maps would increase CCFRL's robustness, ensuring adaptability to diverse carbon intensity patterns and improving its applicability to global federated learning environments.

VII. Conclusion

This research introduced the Carbon-Conscious Federated Reinforcement Learning (CCFRL) framework, a dynamic, dual-objective optimization model that effectively balances carbon efficiency and model performance in federated learning (FL). By leveraging reinforcement learning, CCFRL dynamically adapts client allocation and resource management, outperforming conventional static or heuristic methods. Through advanced state representations, entropy-based client allocation, and adaptive exploration-exploitation strategies, the CCFRL framework demonstrated remarkable improvements in both energy conservation and carbon emission reduction.

The empirical results show that CCFRL achieves significant reductions in carbon emissions—up to 64.23%—and energy consumption—up to 61.78%—without sacrificing model performance, even in the context of non-IID and highly skewed data distributions. The integration of t-test-based stagnation detection, along with the RL-driven adaptive strategies, enabled the system to respond effectively to environmental fluctuations and complex data characteristics. Additionally, CCFRL's ability to dynamically adjust client participation rates led to superior scalability and eco-efficiency, particularly in cloud-edge environments.

In conclusion, CCFRL paves the way for more sustainable and eco-efficient AI practices, aligning decentralized learning with broader sustainability goals. This research highlights the potential of reinforcement learning in balancing immediate resource needs with long-term environmental responsibility, proving that high performance and carbon efficiency can coexist in large-scale, decentralized learning systems. These findings open up new avenues for further research in optimizing resource management across diverse computing platforms, contributing to a greener, more sustainable future in cloud computing and beyond.

Acknowledgment

This research was supported by the Wallenberg AI, Autonomous Systems, and Software Programme (WASP), funded by the Knut and Alice Wallenberg Foundation, and eSSENCE, a strategic research program funded by the Swedish Government. The computations were enabled by the supercomputing resource Berzelius, provided by the National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg Foundation, Sweden. We also thank Peter Münger for his assistance and support at the National Supercomputer Centre of Linköping University.

REFERENCES

[1] M. L. Di Silvestre, S. Favuzza, E. Riva Sanseverino, and G. Zizzo, "How decarbonization, digitalization and decentralization are changing key power infrastructures," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 483–498, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032118304283

[2] A. A. Chien, C. Zhang, L. Lin, and V. Rao, "Beyond PUE: Flexible datacenters empowering the cloud to decarbonize," 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251384627

[3] P. Arroba, R. Buyya, R. Cárdenas, J. L. Risco-Martín, and J. M. Moya, "Sustainable edge computing: Challenges and future directions," *arXiv preprint arXiv:2304.04450*, 2023.

[4] T. Anderson, A. Belay, M. Chowdhury, A. Cidon, and I. Zhang, "Treehouse: A case for carbon-aware datacenter software," 2022.

[5] N. Bashir, D. Irwin, P. Shenoy, and A. Souza, "Sustainable computing-without the hot air," *arXiv preprint arXiv:2207.00081*, 2022.

[6] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Modern Deep Learning Research," in *AAAI Conference on Artificial Intelligence (AAAI)*, February 2020, pp. 13 693–13 696.

[7] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing Carbon: The Elusive Environmental Footprint of Computing," in *HPCA*, February 2021.

[8] Y. Dong, Z. Han, X. Li, S. Ma, F. Gao, and W. Li, "Joint optimal scheduling of renewable energy regional power grid with energy storage system and concentrated solar power plant," *Frontiers in Energy Research*, vol. 10, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenrg.2022.941074

[9] C.-S. Yang, C.-C. Huang-Fu, and I.-K. Fu, "Carbon-neutralized task scheduling for green computing networks," 2022.

[10] F. R. Albogamy, M. Y. I. Paracha, G. Hafeez, I. Khan, S. Murawwat, G. Rukh, S. Khan, and M. U. A. Khan, "Real-time scheduling for optimal energy optimization in smart grid integrated with renewable energy sources," *IEEE Access*, vol. 10, pp. 35 498–35 520, 2022.

[11] G. Perin, F. Meneghello, R. Carli, L. Schenato, and M. Rossi, "Ease: Energy-aware job scheduling for vehicular edge networks with renewable energy resources," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 1, pp. 339–353, 2023.

[12] P. Li, X. Huang, M. Pan, and R. Yu, "Fedgreen: Federated learning with fine-grained gradient compression for green mobile edge computing," pp. 1–6, 2021.

[13] R. Albelaihi, L. Yu, W. D. Craft, X. Sun, C. Wang, and R. Gazda, "Green federated learning via energy-aware client selection," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 13–18.

[14] A. Salh, R. Ngah, L. Audah, K. S. Kim, Q. Abdullah, Y. M. Al-Moliki, K. A. Aljaloud, and H. N. Talib, "Energy-efficient federated learning with resource allocation for green iot edge intelligence in b5g," *IEEE Access*, vol. 11, pp. 16 353–16 367, 2023.

[15] R. Danilak, "Forbes, Why Energy is a Big and Rapidly Growing Problem for Data Centers," https://www.forbes.com/sites/forbestechcouncil/2017/12/15/why-energy-is-a-big-and-rapidly-growing-problem-for-data-centers/?sh=246569c85a30, December 15th 2017.

[16] P. J. Denning and T. G. Lewis, "Exponential Laws of Computing Growth," *Communications of the ACM*, vol. 60, no. 1, pp. 54–65, January 2017.

[17] A. A. Chien, "Driving the cloud to true zero carbon," pp. 5–5, 2021.

[18] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean, "The carbon footprint of machine learning training will plateau, then shrink," *Computer*, vol. 55, no. 7, pp. 18–28, 2022.

[19] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, and I. Sutskever, "AI and Compute," https://openai.com/blog/ai-and-compute/, May 16th 2018.

[20] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurr. Comput.: Pract. Exper.*, vol. 24, no. 13, p. 1397–1420, sep 2012. [Online]. Available: https://doi.org/10.1002/cpe.1867

[21] M. Guazzone, C. Anglano, and M. Canonico, "Energy-efficient resource management for cloud computing infrastructures," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. IEEE, 2011, pp. 424–431.

[22] Y. K. Tun, Y. M. Park, N. H. Tran, W. Saad, S. R. Pandey, and C. S. Hong, "Energy-efficient resource management in UAV-assisted mobile edge computing," *IEEE Communications Letters*, vol. 25, no. 1, pp. 249–253, 2020.

[23] K. Grosskop and J. Visser, "Identification of application-level energy optimizations," *Proceeding of ICT for Sustainability (ICT4S)*, vol. 4, pp. 101–107, 2013.

[24] B. K. Sovacool, "How long will it take? conceptualizing the temporal dynamics of energy transitions," *Energy research & social science*, vol. 13, pp. 202–215, 2016.

[25] F. Creutzig, P. Agoston, J. C. Goldschmidt, G. Luderer, G. Nemet, and R. C. Pietzcker, "The underestimated potential of solar energy to mitigate climate change," *Nature Energy*, vol. 2, no. 9, pp. 1–9, 2017.

[26] A. Grubler, C. Wilson, N. Bento, B. Boza-Kiss, V. Krey, D. L. McCollum, N. D. Rao, K. Riahi, J. Rogelj, S. De Stercke *et al.*, "A low energy demand scenario for meeting the 1.5 c target and sustainable development goals without negative emission technologies," *Nature energy*, vol. 3, no. 6, pp. 515–527, 2018.

[27] Z. Cao, X. Zhou, H. Hu, Z. Wang, and Y. Wen, "Toward a systematic survey for carbon neutral data centers," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 895–936, 2022.

[28] R. Ramesh, Z. Guo, and J. Li, "Study on reducing carbon footprint of data centers," 2023, available online at UBNow: News and views for UB faculty and staff, University at Buffalo. [Online]. Available: https://www.buffalo.edu/ubnow/...

[29] C. Bergaentzlé and T. U. Madsen, "New performance indicators for fully integrated and decarbonised data centres," 2021.

[30] N. Bashir, T. Guo, M. Hajiesmaili, D. Irwin, P. Shenoy, R. Sitaraman, A. Souza, and A. Wierman, "Enabling sustainable clouds: The case for virtualizing the energy system," in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 350–358.

[31] A. Souza, N. Bashir, J. Murillo, W. Hanafy, Q. Liang, D. Irwin, and P. Shenoy, "Ecovisor: A virtual energy system for carbon-efficient applications," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2023, pp. 252–265.

[32] E. Ahvar, S. Ahvar, Z. A. Mann, N. Crespi, J. Garcia-Alfaro, and R. Glitho, "Cacev: A cost and carbon emission-efficient virtual machine placement method for green distributed clouds," in *2016 IEEE International Conference on Services Computing (SCC)*, 2016, pp. 275–282.

[33] B. Wadhwa and A. Verma, "Carbon efficient vm placement and migration technique for green federated cloud datacenters," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2014, pp. 2297–2302.

[34] T. Renugadevi, K. Geetha, N. Prabaharan, and P. Siano, "Carbon-efficient virtual machine placement based on dynamic voltage frequency scaling in geo-distributed cloud data centers," *Applied Sciences*, vol. 10, no. 8, p. 2701, 2020.

[35] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing: 19th International Conference, Aachen, Germany, August 26-30, 2013. Proceedings 19*. Springer, 2013, pp. 317–328.

[36] T. Abbasi-khazaei and M. H. Rezvani, "Energy-aware and carbon-efficient vm placement optimization in cloud datacenters using evolutionary computing methods," *Soft Computing*, vol. 26, no. 18, pp. 9287–9322, 2022.

[37] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig, and D. Rolnick, "Aligning artificial intelligence with climate change mitigation," *Nature Climate Change*, vol. 12, no. 6, pp. 518–527, 2022.

[38] P. Dhar, "The carbon impact of artificial intelligence." *Nat. Mach. Intell.*, vol. 2, no. 8, pp. 423–425, 2020.

[39] H. Hua, H. Wu, J. Shen, K. Li, and Z. Y. Dong, "Machine learning to support low carbon energy transition," *Frontiers in Energy Research*, vol. 11, p. 1175280, 2023.

[40] Z. Yao, Y. Lum, A. Johnston, L. M. Mejia-Mendoza, X. Zhou, Y. Wen, A. Aspuru-Guzik, E. H. Sargent, and Z. W. Seh, "Machine learning for a sustainable energy future," *Nature Reviews Materials*, vol. 8, no. 3, pp. 202–215, 2023.

[41] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[42] X. Qiu, T. Parcollet, J. Fernandez-Marques, P. P. B. de Gusmao, Y. Gao, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, "A first look into the carbon footprint of federated learning," *arXiv preprint arXiv:2102.07627*, 2021.

[43] S. Savazzi, V. Rampa, S. Kianoush, and M. Bennis, "An energy and carbon footprint analysis of distributed and federated learning," *IEEE*

Transactions on Green Communications and Networking, vol. 7, no. 1, pp. 248–264, 2022.
[44] J. Bian, S. Ren, and J. Xu, "Cafe: Carbon-aware federated learning in geographically distributed data centers," *arXiv preprint arXiv:2311.03615*, 2023.
[45] T. Cui, Y. Shi, B. Lv, R. Ding, and X. Li, "Federated learning with sarima-based clustering for carbon emission prediction," *Journal of Cleaner Production*, vol. 426, p. 139069, 2023.
[46] M. S. Al-Abiad, M. Obeed, M. J. Hossain, and A. Chaaban, "Decentralized aggregation for energy-efficient federated learning via d2d communications," *IEEE Transactions on Communications*, vol. 71, no. 6, pp. 3333–3351, 2023.
[47] T. Mehboob, N. Bashir, J. O. Iglesias, M. Zink, and D. Irwin, "Cefl: Carbon-efficient federated learning," *arXiv preprint arXiv:2310.17972*, 2023.
[48] W. E. Armstrong, "The determinateness of the utility function," pp. 453–467, 1939.
[49] E. Seo, D. Niyato, and E. Elmroth, "Resource-efficient federated learning with non-iid data: An auction theoretic approach," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 506–25 524, 2022.
[50] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," 2018.
[51] A. Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[52] E. Seo, D. Niyato, and E. Elmroth, "Auction-based federated learning using software-defined networking for resource efficiency," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 42–48.
[53] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning," *arXiv preprint arXiv:1910.09700*, 2019.
[54] M. Hodak, M. Gorkovenko, and A. Dholakia, "Towards power efficiency in deep learning on data center hardware," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1814–1820.
[55] A. Radovanović, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, S. Talukdar, E. Mullen, K. Smith, M. Cottman, and W. Cirne, "Carbon-aware computing for datacenters," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1270–1280, 2023.

Eunil Seo earned his B.S. degree from the Department of Information Engineering at Sungkyunkwan University, his M.S. degree from the Department of Computer Science at the University of Southern California (USC), and his Ph.D. degree from the Department of Electrical and Computer Engineering at Sungkyunkwan University, in 1998, 2002, and 2019, respectively. Over 27 years, he has built a distinguished career bridging industry and academia, specializing in networking, machine learning, and distributed computing. At Samsung Advanced Institute of Technology (SAIT), he advanced wireless networks, Mobile IP, and IPv6, securing 17 international patents and contributing to the Zigbee network protocol. As Chair of the RIA WG at OMG, he established international standards for Rich Internet Applications. At Umeå University, he has driven innovations in resource-efficient federated learning, IoT, and edge AI, developing methodologies like the Carbon-Conscious Federated Reinforcement Learning (CCFRL) approach to enhance scalability and carbon efficiency in distributed systems. He also led critical technical specifications for the ITER project, integrating advanced solutions in system design and network management.

Erik Elmroth is a Full Professor and has established the Distributed Systems research at Umeå University. He has been head and deputy head of the Department of Computing Science for 13 years and deputy head of a national Supercomputer center for another 13 years. His background covers a broad spectrum of HPC, grid-, and cloud infrastructure research topics. Prof. Elmroth has been Chair of the Swedish National Infrastructure for Computing (SNIC), a member of the Swedish Research Councils Committee for Research Infrastructures, as well as Chairman of its expert group on science infrastructures, and has written two research strategies for the Nordic Council of Ministers. Recognition for his research includes the Nordea Scientific Award and the SIAM Linear Algebra Prize.