Carbon Intensity Forecasting with On-Site Renewables for Sustainable AI Workload Scheduling

Matteo Zanotto*
matteo.zanotto@unitn.it
University of Trento
Trento, Italy

Gabriele Padovani*
gabriele.padovani@unitn.it
University of Trento
Trento, Italy

Giovanni Iacca
giovanni.iacca@unitn.it
University of Trento
Trento, Italy

Sandro Fiore
sandro.fiore@unitn.it
University of Trento
Trento, Italy

Abstract

In recent years, the environmental impact of cloud computing and AI technologies has steadily increased. Data centers currently consume more than 400TWh of electricity each year, and this number is only expected to grow more as companies invest in cloud services and in increasingly compute-hungry Machine Learning (ML) models. While still in an early phase, researchers and industries have started to pay more attention to the rising electricity demand of these technologies, and several computing centers have begun to implement emission reduction measures, such as job execution in low-carbon emissions areas. The allocation of workloads in external regions has, however, led to non-trivial Carbon Intensity (CI) estimations, making it challenging to apply carbon-aware scheduling techniques. Furthermore, accountability measures in these new approaches tend to still be lacking, especially for electricity consumption and emissions, as these metrics are rarely disclosed. This work presents an end-to-end architecture for the minimization of carbon emissions of AI workloads in hybrid cloud environments, based on an ML forecaster, a scheduler, and a provenance-driven accountant component. Experimental results show a 27% reduction in emissions when using the scheduler component against the performance of a carbon-agnostic baseline.

CCS Concepts

• **Computing methodologies** → **Planning and scheduling**; **Machine learning approaches**; • **Information systems** → *Decision support systems.*

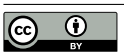
Keywords

Carbon Intensity, Forecasting, Workload Scheduling, Hybrid Cloud, Sustainability

ACM Reference Format:

Matteo Zanotto, Gabriele Padovani, Giovanni Iacca, and Sandro Fiore. 2026. Carbon Intensity Forecasting with On-Site Renewables for Sustainable AI Workload Scheduling. In *2nd International Workshop on Systems and Methods for Sustainable Large-Scale AI (GreenSys) (GreenSys '26), April 27–30, 2026, Edinburgh, Scotland Uk.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3802973.3804455

*Both authors contributed equally to this research.



GreenSys '26, Edinburgh, Scotland Uk

ACM ISBN 979-8-4007-2174-8/2026/04
https://doi.org/10.1145/3802973.3804455

1 Introduction

The rapid expansion of Machine Learning (ML) and large-scale cloud infrastructures has transformed the computational landscape, leading to unprecedented advancements in several scientific fields. However, this growth has come at a high environmental cost. Data centers alone already account for nearly 415TWh of electricity usage (~1.5% of global availability) [23], and this figure is projected to grow to nearly 945TWh by 2030, as increasingly complex ML models demand vast computational and energy resources [10]. To this end, estimating the environmental footprint of computation has emerged as an increasingly pressing research challenge.

In response, data center operators and cloud providers have begun to integrate on-site renewable and low-carbon energy sources – such as solar, wind, and nuclear – into their power supply mix. This shift represents an important step towards sustainable computing, but it also introduces new challenges in accurately estimating the emissions of computations using Carbon Intensity (CI), which is essential for applying emerging emission-aware workload scheduling techniques. Furthermore, cloud providers rarely disclose metrics on the energy consumption and emissions caused by the execution of workloads [14, 18]. This lack of transparency makes it challenging for end users to make informed decisions about environmentally aware scheduling of compute resources. This issue is further complicated by the widespread adoption of hybrid cloud and federated computing environments that feature intricate scheduling and management decisions [3]. Accurate estimations of CI and transparent management of resources are therefore required for the sustainability of cloud environments.

This work presents an end-to-end architecture for carbon-aware scheduling in multi-cloud environments. This system makes use of forecasted CI from several geographical regions to instruct a scheduler that dynamically allocates compute-oriented workloads based on energy availability, predicted emissions, and resource constraints. Given multiple geographically distributed clusters with distinct energy availability and CI profiles, the scheduler can either select the optimal execution zone for each job based on CI and

resource availability or delay job execution when forecasts predict higher renewable availability and lower CI in the near future. The proposed approach, additionally, implements a downstream accounting system based on provenance documents, with the intent of making the process transparent from a user perspective.

The rest of the paper is organized as follows. The next section briefly overviews the related work. Section 3 describes the proposed methodology, followed by Section 4 that presents the results. Finally, Section 5 provides the conclusions.

2 Related Work

In recent years, interdisciplinary research initiatives at the intersection of energy systems, data science, and environmental policy have received visible interest, reflecting the growing imperative to address climate change through enhanced energy management.

A significant portion of existing literature focuses on real-time energy monitoring systems and carbon accounting tools, many of which leverage open datasets and visualization platforms to enhance transparency and public awareness [6]. Electricity Maps (EMaps)[1] stands out among these efforts for its intuitive interface and emphasis on CI metrics. However, similar platforms, such as WattTime[2] [24] and CI API[3] also offer insights into real-time emissions, with features for automated emissions reduction through demand response. While these tools provide valuable services, most remain limited in geographic scope and resolution [9], highlighting the need for optimization approaches that fine-tune and model the data for more accurate reconstruction.

Nowadays, the data made available by these systems can be handled by a number of ML techniques, such as foundation models (FMs), increasing the number of competitive models designed for general-purpose time series forecasting. Many of these models can also be pretrained on large historical datasets and be used in a zero-shot manner, without the need to finetune. Among the more accurate ones, the majority are based on FMs [2, 7, 11], or LLMs [30], and only a few rely on smaller models with a focus on scalability [8, 16]. Only very recently, with ever more attention being paid to carbon footprint and impact of ML training on the environment, AI researchers have focused on reducing the parameter number of these models[4], favoring ones with comparable performance [1].

Leveraging the availability of data regarding real-time electricity monitoring and the ability to forecast such information through ML techniques, the carbon-aware paradigm has emerged as a valid approach toward reducing the emissions of cloud computations [21]. Specifically, it consists of aligning the execution of workloads according to the availability of renewable energy, which by nature is volatile and fluctuates over time depending on seasonal and geographical differences that affect its production. In the literature, various architectures [12, 19, 25, 26] have been proposed to exploit temporal variations by delaying time-insensitive jobs to periods with ample green energy availability. Similarly, multiple approaches [4, 20, 27] have been proposed to reduce computation emissions by exploiting geographical variations in CI, offloading workloads between geographically distributed clusters. Various architectures have also investigated the joint effects of time- and space-shifting in scheduling of cloud workloads [5, 29].

Current works considering hybrid cloud scenarios optimize scheduling to address aspects related to cost efficiency, with constraints on Quality of Service (QoS) or security conditions [15, 22, 31, 32], while the sustainability perspective has not yet been addressed.

This work aims at exploiting ML-based forecasters for CI and on-site power generation to apply established emission-aware scheduling approaches for minimizing workloads' emissions in a hybrid-cloud environment, producing provenance documents for transparently describing resource management decisions.

3 Methodology

3.1 System Model

This study proposes a system in which users (e.g., ML researchers) submit jobs to their organization's computing infrastructure, aiming at minimizing the environmental impact of its services.

The organization manages a hybrid cloud computing infrastructure composed of a private local cluster and a set of geographically distributed public cloud computing services hosted by Cloud Service Providers (CSPs). Let $J = \{0, 1, \ldots, J_{max}\}$ denote the set of clusters with $j \in J$ identifying a cluster and $j = l$ with $l = 0$ being the local cluster. Each cluster operates in a region with carbon intensity $CI_j(t)$ at time t. The local cluster draws power from its regional grid and from on-site renewable generation (e.g., solar panels). Its adjusted CI is denoted as $CI_l(t)$ and detailed further on. The local cluster consists of on-premise nodes, with N^{max} being the maximum number of reservable nodes. The public cloud computing services, instead, are accessed via subscription plans with one or more CSPs to provide additional capacity when the local cluster is saturated, and to enable carbon-aware scheduling. These subscription plans impose Service Level Agreements (SLAs), with Q_j^{max} denoting the usage quota limit at cluster j (the maximum number of available nodes reservable by workloads at cluster j).

Workloads are modeled as non-preemptive tasks that run to completion once started. Each workload executes on the N requested nodes of a single cluster, with exclusive node reservation. Furthermore, workloads are assumed to have uniform computational load and constant power consumption (such as in the case of ML training jobs). Workloads are scheduled upon arrival without being queued, although their execution start may be delayed by the scheduler.

Considering these assumptions, a workload is defined as $W := < D, N >$, where D is the user-specified duration (wall-time) and N is the number of required nodes.

The aim of the proposed solution is to optimally allocate workloads so that the local cluster usage is maximized, while the computing-related emissions are minimized. This architecture acts as a broker between job submission and execution. To ensure accountability during the whole process, a provenance-based tracker records each scheduling decision, pairing it with contextual information. A high-level diagram of the architecture is shown in Figure 1.

The proposed architecture manages the ML workloads submitted by users in a pipeline consisting of three main components: a Forecasting module, which predicts CI and power generation in a specific zone; a Scheduler, which selects the time and cluster in

[1] https://app.electricitymaps.com/map/72h/hourly
[2] https://watttime.org/
[3] https://www.carbonintensity.org.uk/
[4] https://research.google/blog/a-decoder-only-foundation-model-for-time-series-forecasting/

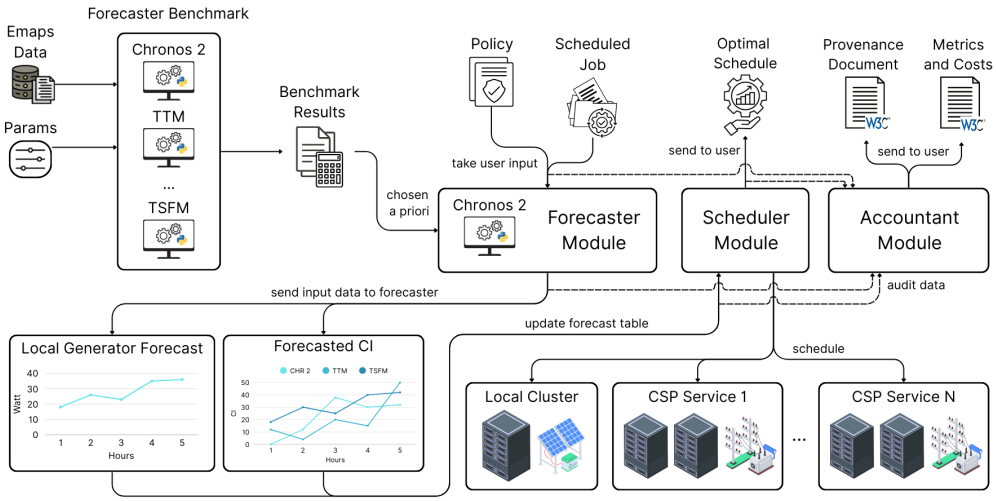


Figure 1: Summary of the scenario addressed in this work. Users submit jobs alongside Scheduler parameters. The system relies on three components, Forecaster, Scheduler, and Accountant, to optimally schedule the job and return the result.

which to execute the workload; and an Accountant, which tracks the scheduling decisions and execution metrics. For the first module, a set of zero-shot forecasters is used to predict the CI at an arbitrary time window of a zone of the world. Such forecasts are cached for a configurable time validity for re-use by the Scheduler to avoid additional computations. The CI of the local cluster $CI_l(t)$ at time t is determined when admitting a user workload $W =< N, D >$ by considering how much of the power demand of the cluster is met with on-site renewable energy generation. The power produced by the local generator $P_{gen}(t)$ at time t is forecasted based on historical data, analogously to the CI predictions. We assume such power to be fully utilized by the cluster, with excess demand met with power drawn from the grid, as given by:

$$\%P_{grid}(t) = \frac{min\{(C_{hw} \cdot (N + N_l(t))) - P_{gen}(t), 0\}}{C_{hw} \cdot (N + N_l(t))}. \quad (1)$$

where $C_{hw} \cdot (N + N_l(t))$ is the electricity demand of the cluster considering the allocation of the workload's requested nodes N and the nodes already allocated at time t given by $N_l(t)$, multiplying by the node's hardware power consumption requirement (C_{hw}). By dividing by the total power demand, a percentage measure is obtained. This way, the adjusted CI is computed by weighting the generator's CI and the grid's CI depending on their percentages of the cluster power demand:

$$CI_l(t) = \%P_{grid}(t) \cdot CI_j(t) + \left(1 - \%P_{grid}(t)\right) \cdot EF_{gen} \quad (2)$$

where $CI_j(t)$ is the CI in gCO_2eq/kWh of the grid underlying the local cluster at region j at time t and EF_{gen} is the emission factor in gCO_2eq/kWh of the electricity source used by the local generator. This adjustment can be seen in Figure 2.

In our experiments, the Scheduler and forecasters were tested on eight different regions: Australia (AU-NSW), Germany (DE), the United Kingdom (GB), Italy (IT-NO), Japan (JP), Korea (KR), the United States ERCOT (US-TEX-ERCO), and the United States PJM

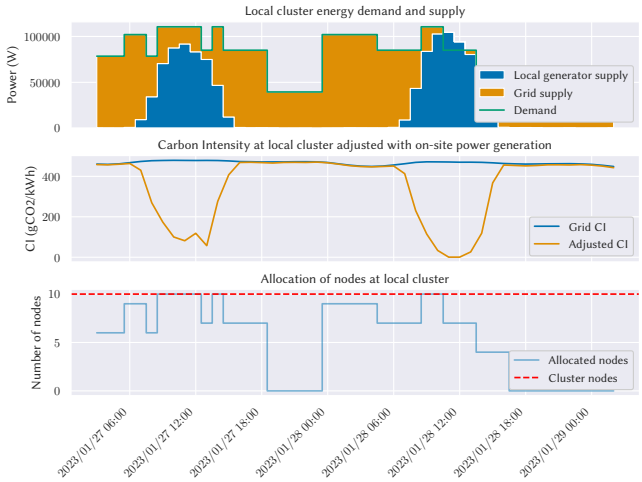


Figure 2: From the top: the local cluster power demand and supply according to Chronos 2; the resulting CI when adjusted, taking into account the local generator; and the number of scheduled jobs in the local cluster over time.

(US-MIDA-PJM). These regions were selected based on infrastructural and seasonality differences. To have ground truth information on all the forecasters, we selected the year 2023 from EMaps data, since it is the most recent year to include all the required variables. Due to page limitations, the accounting system section has been summarized; future work will focus on a more in-depth overview of this module.

3.2 Forecasting

To perform the forecasting for hourly CI, a set of time-series ML models was benchmarked. Given the relatively small amount of

data present in the EMaps dataset, it was decided to rely on zero-shot inference from already pretrained models. All forecasters were able to run on a single NVIDIA A100 GPU with 32 GB of RAM.

The ML architectures selected are: Tiny Time Mixers [8], a pre-trained model designed for multivariate time-series; Time Series FM [7], a decoder-only FM; Moment FM [11], an FM designed for general-purpose time-series; Chronos [2] and Chronos 2 [1], adapted transformer-based LLMs for probabilistic time series; and GPT4TS [30], another LLM for time series. In addition to this two baseline models were defined, a naïve approach that outputs the most recent 96-hour sequence it has access to, which is referenced as "Naïve Repeat", and an ARIMA [17] based model. CarbonCast [16] and EnsembleCI [28] were initially taken into consideration, but discarded due to the multi-variate nature of the former and an incompatibility in the testing architecture for the latter. All forecasters were tested in a zero-shot setting on all eight zones by reproducing the entire 2023 year of data, and Mean Squared Error (MSE) was used as a metric to evaluate each model's performance. Each model was given a context of 512 hourly timesteps, amounting to 21 days of information, and was tasked with inferring the next 96 timesteps. These values were chosen since they are the maximum context sizes, both for the input and the output, common to each Forecaster.

3.3 Scheduling

A Linear Programming (LP) model is defined to implement the scheduling procedure that allocates workloads to (1) minimize the emissions resulting from their execution and (2) maximize the usage of the local computing cluster.

Some preliminary definitions are introduced below. Let W be the workload to be allocated. Let J be the set of eligible scheduling clusters, indexing the CI forecasts for each region as described before. In particular, public cloud clusters are denoted as j while the local on-premise cluster is encoded as l, with $j, l \in J$ and $l = 0$. Let $t \in [0, T]$ be the hourly time-steps of the scheduling horizon of length T. Finally, let x_{tj} be a binary variable equal to one if the workload is allocated at cluster j at time t; or zero otherwise. Two objective functions can thus be defined.

(1) Emissions: Starting a workload at time t at cluster j results in the following emissions:

$$E(t, j) = x_{tj} \cdot c_{tj} \tag{3}$$

where c_{tj} is the CI measured at cluster j at time t for the duration D of the workload, which is given by:

$$c_{tj} = \sum_{\tau=t}^{t+D} CI_j(\tau).$$

The Forecaster component described earlier is used to build the CI data matrix. More specifically, the CI of the local cluster CI_l contains the adjusted CI from the power mix of the on-site generator and the CI from the excess electricity demand consumed from the grid.

It should be noted that many public cloud providers also use their own on-site power generators and therefore often only partially rely on grid electricity, much like the local cluster modeled thus far. Consequently, estimating the emissions of computations executed at their data centers using the carbon intensity of their underlying grid may not be fully representative. This modeling issue is shared by many region-based, carbon-aware workload scheduling approaches. One solution to this problem would be to use data about emissions or renewable power production directly released by cloud providers, which they publish very sparingly [3, 18]. Nevertheless, the proposed methodology of adjusting the local carbon intensity depending on on-site generation is a flexible approach that can also be applied to public cloud services, provided data on their renewable power production is made available.

(2) Usage of local computing cluster: Starting a workload at time t at cluster j results in the following usage of local cluster nodes:

$$L(t, j) = x_{tj} \cdot a_{tj} \tag{4}$$

where a_{tj} determines the number of nodes allocated at the local cluster l for the task, which is given by:

$$a_{tj} = \begin{cases} N, & \text{if } j = l, \\ 0, & \text{otherwise.} \end{cases}$$

This objective function also approximates a measure of the expenses of the cluster operator for managing the computing infrastructure, with inverse proportionality. This is because maximizing the usage of the local cluster will contribute to amortizing the costs for hardware purchase and infrastructure setup, while reducing the expenses for public cloud services due to the pay-per-use business model adopted by the cloud computing paradigm. As such, maximizing the usage of the local cluster indirectly also minimizes the overall expenses of the cluster operator.

The Scheduler component implements a trade-off between the minimization of the workload emissions (1) and the maximization of the local cluster usage (2) by jointly optimizing the cost functions:

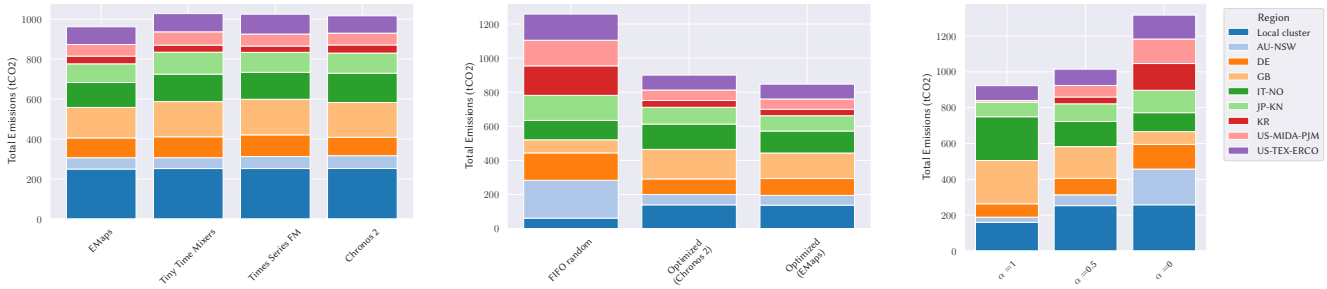
$$t^*, j^* = \text{argmin}_{t,j}\{\alpha \bar{E}(t, j) - (1 - \alpha)\bar{L}(t, j)\} \tag{5}$$

where: $\alpha \in [0, 1]$ is a weighting factor used to control the trade-off between emissions and local cluster usage; and $\bar{E}(t, j)$ and $\bar{L}(t, j)$ are the CI and local cluster usage cost functions previously defined in Equations (3) and (4) respectively, but normalized over their maximum value. This is done to make sure they both return values $\in [0, 1]$ and thus are comparable on the same scale.

To respect the Service Level Agreement (SLA) on job allocation and to ensure the feasibility of the solution, the following constraints must hold:

(1) $\sum_t^T \sum_j^J x_{tj} = 1$, meaning workloads are allocated exactly once at a certain time and region;
(2) $t + D <= T$, meaning the workload must be completed before the scheduling horizon T, considering its wall-time D;
(3) $\forall \tau \in [t, t + D],\ N + N_l(\tau) \leq N^{max}$ meaning the number of nodes allocated at the local cluster l at time τ (stored in $N_l(\tau)$) cannot exceed N^{max}, which is the maximum number of nodes available for the duration of the workload;
(4) $\forall \tau \in [t, t + D],\ N + N_j(\tau) \leq Q_j^{max}\ \forall\ j \in J, j \neq l$ meaning the number of nodes allocated at CSP cluster j at time τ (stored in $N_j(\tau)$) cannot exceed the usage quota limit Q_j^{max} for the duration of the workload to schedule.

Being a linear formulation with linear constraints, the resulting optimization problem can be efficiently solved using available linear solvers. Furthermore, the solution will always be the global



(a) Performance comparison of this approach when using the three best forecasting models.

(b) Performance comparison of this approach against a carbon-agnostic FIFO scheduler.

(c) Impact of the α trade-off parameter used in Equation (5).

Figure 3: A summary of the experimental section of this work. The contributions to the total emissions of workloads executed at each region are highlighted using the same color.

Table 1: Summary of the models and baselines used, and zero-shot performance comparison on CI prediction.

Model	Energy (mWh)	MSE Loss	Params
Naïve Repeat	0.0025	1218.47	-
ARIMA	0.4444	2876.42	-
Moment FM	2.7569	498.63	347M
Tiny Time Mixers	1.2569	198.36	800k
Chronos	8.3542	362.74	46M
Chronos 2	**7.2431**	**176.58**	**120M**
Time Series FM	6.8333	243.91	231M
GPT4TS	2.0625	547.82	65M

optimum as the constraints are convex. In our experimentation, we used the PuLP library[5], and solved it with the Cbc solver[6].

4 Results

4.1 Forecaster Performance

Table 1 contains a recap of all the models utilized, with the respective performance, number of parameters, and inference time. Due to the page limit constraints, the benchmark for this initial phase has been summarized, and further information will be provided in an extension to this work. We initially benchmarked the predictions made by the different zero-shot models by comparing them with the Electricity Maps (EMaps) ground truth. Across all regions, both baselines ('Naïve Repeat" and ARIMA) consistently exhibit the highest MSE. In contrast, all evaluated ML models achieve lower test MSE, with Chronos 2 being the best, followed by TSFM and TTM. Since the top three models all achieved relatively similar performance, it was decided to compare the scheduling performance based on all three architectures, as shown in Figure 3a.

4.2 Scheduler Performance

The Scheduler component was evaluated with multiple experiments conducted over various configurations. More specifically, simulations of job scheduling were conducted with a total of 1000 jobs arriving per month. The characteristics of the submitted jobs were generated using a uniform distribution over the whole period and generating the number of requested nodes N and wall-time D (in hours) using normal distributions with parameters $\mu = 6$ and $\sigma = 2$.

A hybrid cloud architecture was simulated considering a local cluster equipped with a solar power generator and a set of CSP clusters located across the eight above-mentioned regions. In this simulation, the local cluster provides an N^{max} of 10 nodes, while the on-site generator is modeled using EMaps data on solar power production, ensuring a realistic output depending on seasonal variations. The CSP clusters instead provide a usage quota limit of $Q_j^{max} = 20 \; \forall j$.

Nodes executing allocated workloads were assumed to have a constant power demand throughout the job's duration, set as $C_{hw} = 8.5kW$ as measured in [13].As such, the emissions caused by a workload $W :=< D, N >$, starting at time t in cluster j, with $CI_j(\tau)$ being the CI at region j at time τ were computed as:

$$Em(D, N, t, j) = \sum_{\tau=t}^{t+D} N \cdot C_{hw} \cdot CI_j(\tau)$$

The performance of the proposed carbon-aware computing approach was evaluated against a random FIFO baseline scheduler, which assigns workloads to randomly chosen regions, respecting resource constraints. The scheduler was evaluated considering both forecaster data (from Chronos 2) and ground truth EMaps data for a theoretical upper bound in the achievable reduction in emissions. As shown in Figure 3b, the optimized model reduces emissions by 27.7% compared to the random FIFO scheduler, while a reduction of 32.09% can be achieved using ground truth knowledge.

The impact of the trade-off parameter α was also evaluated, considering three configurations corresponding to different scheduling profiles. With $\alpha = 1$, the model minimizes overall emissions regardless of local cluster usage. Conversely, with $\alpha = 0$, the model maximizes local cluster usage without accounting for emissions, as by Equation (4). With $\alpha = 0.5$, a tradeoff is adopted between maximizing local cluster usage and minimizing emissions. The results are shown in Figure 3c, where the best tradeoff allocates the majority of jobs to the local cluster with an increase of only 9.97% of emissions compared to the $\alpha = 1$ profile.

[5]https://coin-or.github.io/pulp/
[6]https://github.com/coin-or/Cbc

5 Conclusions

The methodology proposed in this work integrates power consumption modeling and CI estimation into a unified framework for sustainable job allocation in hybrid cloud scenarios.

A benchmark of state-of-the-art foundation models for time-series forecasting was conducted to evaluate the performance of existing approaches, taking into account performance, energy consumption, and inference time of each model. The best forecasters were then used to predict the CI at different regions and the on-site renewable power generation, for the adjustment of local CI.

A Linear Programming optimization model was defined for the carbon-aware scheduling of jobs in hybrid cloud scenarios. The model computes a controllable trade-off between the minimization of the carbon footprint of computations and the maximization of local cluster usage.Due to page limit constraints, the experimental evaluation had to be shortened, but additional work has been conducted analyzing the scalability of this approach, its effectiveness when increasing the delay time window, and when using coarser time series data. Future research efforts plan to relax the assumption of constant power draw of workloads. Furthermore, the model will be extended to take into account different types of jobs (GPU-bound, IO-bound, etc.) and nodes (hardware characteristics). Finally, a provenance-driven accounting system was implemented to enable assessments about the management of workloads; more details on this component will be disclosed in a future extension of this work.

Acknowledgments

This work was funded by the European Union – Next Generation EU, Mission 4 Component 1 CUP E66E24000060008.

References

[1] Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. 2025. Chronos-2: From Univariate to Universal Forecasting. arXiv preprint arXiv:2510.15821.

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. arXiv preprint arXiv:2403.07815.

[3] Rohan Arora, Umamaheswari Devi, Tamar Eilam, Aanchal Goyal, Chandra Narayanaswami, and Pritish Parida. 2023. Towards carbon footprint management in hybrid multicloud. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. ACM, New York, NY, USA, 1–7.

[4] Tayebeh Bahreini, Asser Tantawi, and Alaa Youssef. 2023. A carbon-aware workload dispatcher in cloud computing systems. In *2023 IEEE 16th International Conference on Cloud Computing*. IEEE, New York, NY, USA, 212–218.

[5] Tayebeh Bahreini, Asser N Tantawi, and Olivier Tardieu. 2024. Caspian: A Carbon-aware Workload Scheduler in Multi-Cluster Kubernetes Environments. In *2024 32nd International Conference on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE, New York, NY, USA, 1–8.

[6] Olivier Corradi. 2025. Mapping global electricity emissions in real time. *Nature Reviews Clean Technology* 1 (2025), 1–2.

[7] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*. JMLR.org, Vienna, Austria, 20 pages.

[8] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. 2024. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems* 37 (2024), 74147–74181.

[9] Ember. 2025. Ember Energy Report. Retrieved from https://ember-energy.org/latest-insights/global-electricity-review-2025/.

[10] Daniel Raphael Ejike Ewim, Nwakamma Ninduwezuor-Ehiobu, Ochuko Felix Orikpete, Blessed Afeyokalo Egbokhaebho, Akeeb Adepoju Fawole, and Chiemela Onunka. 2023. Impact of data centers on climate change: a review of energy efficient strategies. *The Journal of Engineering and Exact Sciences* 9, 6 (2023), 16397–01e.

[11] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. arXiv preprint arXiv:2402.03885.

[12] Walid A Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. 2024. Going green for less green: Optimizing the cost of reducing cloud carbon emissions. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*. 479–496.

[13] Imran Latif, Alex C Newkirk, Matthew R Carbone, Arslan Munir, Yuewei Lin, Jonathan Koomey, Xi Yu, and Zhiuha Dong. 2024. Empirical Measurements of AI Training Power Demand on a GPU-Accelerated Node. arXiv preprint arXiv:2412.08602.

[14] Lawrence Berkeley National Laboratory. 2024. *United States Data Center Energy Usage Report*. Technical Report. Lawrence Berkeley National Laboratory. https://eta-publications.lbl.gov/sites/default/files/2024-12/lbnl-2024-united-states-data-center-energy-usage-report.pdf Accessed: 2026-03-17.

[15] Jian Lei, Quanwang Wu, and Jin Xu. 2022. Privacy and security-aware workflow scheduling in a hybrid cloud. *Future Generation Computer Systems* 131 (2022), 269–278.

[16] Diptyaroop Maji, Prashant Shenoy, and Ramesh K Sitaraman. 2022. CarbonCast: multi-day forecasting of grid carbon intensity. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. ACM, New York, NY, USA, 198–207.

[17] Shahenaz Mulla, Chaitanya B Pande, and Sudhir K Singh. 2024. Times series forecasting of monthly rainfall using seasonal auto regressive integrated moving average with EXogenous variables (SARIMAX) model. *Water Resources Management* 38, 6 (2024), 1825–1846.

[18] David Mytton. 2020. Hiding greenhouse gas emissions in the cloud. *Nature Climate Change* 10, 8 (2020), 701–701.

[19] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. 2022. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems* 38, 2 (2022), 1270–1280.

[20] Abel Souza, Shruti Jasoria, Basundhara Chakrabarty, Alexander Bridgwater, Axel Lundberg, Filip Skogh, Ahmed Ali-Eldin, David Irwin, and Prashant Shenoy. 2023. Casper: Carbon-aware scheduling and provisioning for distributed web services. In *Proceedings of the 14th International Green and Sustainable Computing Conference*. ACM, New York, NY, USA, 67–73.

[21] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. Spatiotemporal Carbon-aware Scheduling in the Cloud: Limits and Benefits. In *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems*. ACM, New York, NY, USA, 2 pages.

[22] Zaixing Sun, Hejiao Huang, Zhikai Li, Chonglin Gu, Ruitao Xie, and Bin Qian. 2023. Efficient, economical and energy-saving multi-workflow scheduling in hybrid cloud. *Expert Systems with Applications* 228 (2023), 120401.

[23] Diana Ürge-Vorsatz and Felix Creutzig. 2026. Energy demand and decarbonization in 2025 and beyond. *Nature Reviews Clean Technology* 2, 1 (2026), 4–5.

[24] WattTime. 2021. WattTime. Retrieved from https://www. watttime. org/.

[25] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. 2021. Let's wait awhile: How temporal workload shifting can reduce carbon emissions in the cloud. In *Proceedings of the 22nd International Middleware Conference*. ACM, New York, NY, USA, 260–272.

[26] Kaiqiang Xu, Decang Sun, Han Tian, Junxue Zhang, and Kai Chen. 2025. {GREEN}: Carbon-efficient Resource Scheduling for Machine Learning Clusters. In *22nd USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association, Berkeley, CA, USA, 999–1014.

[27] Minxian Xu and Rajkumar Buyya. 2020. Managing renewable energy and carbon footprint in multi-cloud computing environments. *J. Parallel and Distrib. Comput.* 135 (2020), 191–202.

[28] Leyi Yan, Linda Wang, Sihang Liu, and Yi Ding. 2025. EnsembleCI: Ensemble Learning for Carbon Intensity Forecasting. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems (e-Energy)*. doi:10.1145/3679240.3734630

[29] Daming Zhao and Jiantao Zhou. 2022. An energy and carbon-aware algorithm for renewable energy usage maximization in distributed cloud data centers. *J. Parallel and Distrib. Comput.* 165 (2022), 156–166.

[30] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. *Advances in Neural Information Processing Systems* 36 (2023), 43322–43355.

[31] Jie Zhu, Xiaoping Li, Rubén Ruiz, and Xiaolong Xu. 2018. Scheduling stochastic multi-stage jobs to elastic hybrid cloud resources. *IEEE Transactions on Parallel and Distributed Systems* 29, 6 (2018), 1401–1415.

[32] Liyun Zuo, Lei Shu, Shoubin Dong, Yuanfang Chen, and Li Yan. 2016. A multi-objective hybrid cloud resource scheduling method based on deadline and cost constraints. *IEEE access* 5 (2016), 22067–22080.