CarbonOracle: Automating Energy Mix & Renewable Energy Source Forecast Modeling for Carbon-Aware Micro Data Centers

Moysis Symeonides*, Nicoletta Tsiopani*, Georgios Maouris*, Demetris Trihinas†, George Pallis*, Marios D. Dikaiakos*

* Department of Computer Science
University of Cyprus
Email: {msymeo03, ntsiop01, gmaour01, pallis, mdd}@ucy.ac.cy

† Department of Computer Science
University of Nicosia
Email: trihinas.d@unic.ac.cy

***Abstract*—Geo-distributed data centers (DCs) have a substantial impact on global electricity consumption and carbon emissions, with their energy demands expected to increase alongside emerging technologies such as Generative Artificial Intelligence (GenAI) and Natural Language Understanding (NLU). In response to environmental and operational concerns, major cloud providers are investing in DC infrastructures powered by renewable energy sources (RES). However, the design and management of energy-efficient data centers present new challenges. Current forecasting models for RES production and electricity grid energy mix are often limited in accuracy and forecasting horizon, hindering carbon-aware service management. To tackle these challenges, we introduce CarbonOracle, a Machine Learning (ML) service that automates data extraction from self-hosted RES, energy grids, and weather APIs, while also simplifying the ML training and forecast of RES production and electricity grid carbon emissions. Its application programming interface serves ML-based forecasts for RES production (e.g., solar, wind) and energy mix metrics, designed to support carbon-aware deployments, enabling integration with container schedulers and other applications. Through a comprehensive evaluation over a real data center testbed, our results show that CarbonOracle has an error rate of approximately 9% for forecasts related to self-hosted photovoltaic (PV) panels, while its forecasts for electricity grid carbon emissions have an error rate of less than 4%.**

***Index Terms*—Data centers, Energy Modeling, Sustainable Computing, Machine Learning.**

I. Introduction

Geo-distributed data centers (DCs) require vast amounts of electricity to operate. Recent studies estimate that data centers consumed between 240 and 340 TWh of electricity in 2022, representing roughly 1-1.3% of the global electricity demand [1], [2]. This vast energy toll not only increases the operational costs of DC owners but also negatively contributes to global carbon emissions. In 2020 alone, DCs were responsible for emitting approximately 330 Million metric tons of CO_2 [2], and projections indicate this figure could rise up to 2.5 Billion tons by 2030, driven by the growing electricity demands for Generative AI [3]. Current trends reflect this increase, with Google DCs seeing a 48% rise in CO_2 emissions since 2019 [4]. This situation exemplifies the growing pressure on tech companies to balance digital expansion with sustainable computing practices, in line with initiatives such as the European Green Deal [5] and the UK net-zero policies [6]. Major cloud providers, such as Google, AWS, and Microsoft, have promised to deliver carbon-neutral services in the coming years [7]–[9], by investing in greener infrastructure powered by renewable energy sources (RES).

The latter brings new challenges in the design, implementation, and operation of data centers [10]. Targeting the minimization of carbon emissions, researchers propose "smart" workload scheduling and placement on the Edge-Cloud Computing continuum [11]–[13]. These systems organize the workload execution so that during periods where the electricity grid is using low carbon energy sources [12], such as during windy weather that boosts energy production from wind turbines, or executing the workload when self-hosted renewable energy production is higher, like a solar-powered data center being most efficient during sunny middays [13]. Operators may also manage multiple micro-DCs distributed across different locations—ranging from cities to countries or continents. In these geo-distributed DCs, energy production from RES depends on regional weather patterns and the local day-night cycle, while each host nation's energy mix influences overall carbon emissions [11]. In these cases, long-term and accurate forecasts are critical for both RES-equipped geo-distributed DCs and grid energy mix to provide more effective service management.

Unfortunately, current solutions fall short when providing long-term predictions. On the one hand, general models for RES production need as input the current and forecast weather conditions along with a large number of parameters that describe the RES installation. For example, PV forecast models need the size of the PVs, their orientation and installation angles, and the level of dust on them, among others [14] [15]. It is almost impossible for the average researcher to know these details, so the predictions of the latter models introduce a notable error. In addition, the latter models cannot capture fine-grained properties of self-hosted RES, like cumulative failures and performance degradation. On the other hand, there are many services providing metrics for the grid energy mix of various countries, like ENTSO-e [16] or Electricity Maps [17] , but with no or limited forecast horizon (i.e., max 24 hours). This situation drives researchers that build carbon-aware services, either to create their own forecast models, which introduces a high learning curve related to AI/ML methods and increases the implementation duration, or to use historical data, assuming their values are known [12].

To tackle these problems, we introduce CarbonOracle, an automated ML service for modeling grid energy mix and RES to facilitate carbon-aware deployments and algorithms. Specifically, CarbonOracle service: (i) introduces an automated process for data extraction and scraping of self-hosted RES, grid energy mix online sites, and weather data from APIs; (ii) abstracts the AI/ML forecast training methodology and materializes the latter methodology into two training pipelines, having as targets the production values of RES and the energy mix metrics, namely, PV and wind production levels, overall grid energy consumption, and country's forecast carbon emissions; and (iii) offers a high-level RESTful application programming interface (API), which allows researchers and programs, like container schedulers (e.g., Kubernetes), to retrieve forecasting data. We have evaluated our methodology with real-world data extracted from three self-hosted RES photovoltaic (PV) panel racks [10] and on energy grid data from Cyprus [18]. Our results indicate that CarbonOracle achieves high performance with a small error, about 9%, for PV panels, while Energy Mix modeling predicts the carbon emissions of the country with an error of about 4%.

The rest of the paper is structured as follows: Sec. II introduces a motivating example about the topic. Next, Sec. III and IV show the system overview and its details. In Sec. V, we evaluate the AI/ML models, while Sec. VI and VII present the related work and conclude the paper, respectively.

II. Motivating Example

To illustrate the practical application of CarbonOracle, consider a use case involving a company focusing on AI services that operate on self-hosted cloud infrastructures (micro-DCs) across multiple regions. Given the energy demands related to the training of deep learning (DL) models, such as large language models (LLMs), and the company's dedication to environmental neutrality, its micro-DCs are equipped with RES. The company seeks to minimize its overall carbon emissions by creating an autonomous system that determines "when" and "where" the ML training and inference workloads should be executed, reducing its carbon footprint. This autonomous system requires accurate forecasts of energy production from the company's RES, as well as predictions of the electricity grid's carbon intensity. However, the company's engineers lack the expertise to build these forecasting models, and existing online services do not provide long-term predictions for carbon intensity across different regions.

To address these challenges, engineers working on carbon-aware systems can leverage CarbonOracle, an automated ML service specifically designed for forecasting renewable energy production and grid carbon emissions. CarbonOracle simplifies the process by automatically collecting the necessary data and building and storing the required ML models for accurate forecasting. Through its API, users can seamlessly retrieve predictions for their renewable energy sources and grid carbon intensity, integrating this information into their auto-scheduling and workload placement systems. Additionally, users can easily add new data sources to account for future RES deployments or grid emissions from new locations.

III. The CarbonOracle Service

A. Features & Objectives

To facilitate advanced carbon-aware service orchestrators that operate on modern computing paradigms such as RES-enabled geo-distributed DCs, CarbonOracle is designed with the following **key features**:

- **Automated Model Generation:** automates the creation and updating of forecasting models for renewable energy sources (RES) and national energy mixes, tailored to user-defined configurations.
- **Seamless Data Integration:** allows the system to efficiently gather data from diverse sources, including weather APIs, energy production platforms, and RES monitors, to perform model training.
- **Real-time Inference via API:** provides a set of RESTful API endpoints that allow users and systems to access real-time forecasts of energy production and carbon emissions.
- **Configurable and Customizable Instantiation:** enables users to adjust system parameters, such as geographic locations and training schedules, and create custom forecasting models for various energy sources.
- **Extensibility and Ease of Integration:** facilitate seamless integration with minimal coding effort through clearly defined programming interfaces, allowing users to expand and adapt the service.

The primary **objective** of CarbonOracle is to offer a service that *surpasses the current State-of-the-Art by automating the (re-) training, storing, and serving of RES and energy mix forecast models, which are critical for carbon-aware scheduling algorithms within orchestration frameworks*.

B. CarbonOracle Overview

Figure 1 depicts a high-level overview of the CarbonOracle service. The onboarding starts with users submitting a set of configurations, which include the geo-location of the underlying renewable sources, training periodicity, training and forecasting parameters, etc. The system will parse the respective parameters and configure the underlying service processes. Specifically, CarbonOracle introduces two processes, namely: (i) *Forecasting Training*, which is executed periodically and asynchronously to create and update the forecasting models; and (ii) *Forecasting Inference* through which the system serves its predictions via its RESTful API.

The Forecasting Training process can be triggered either from an API request or via *Recurrent Triggering*, e.g., once per week, configured through the submitted configurations. When the Forecasting Training process is triggered, CarbonOracle

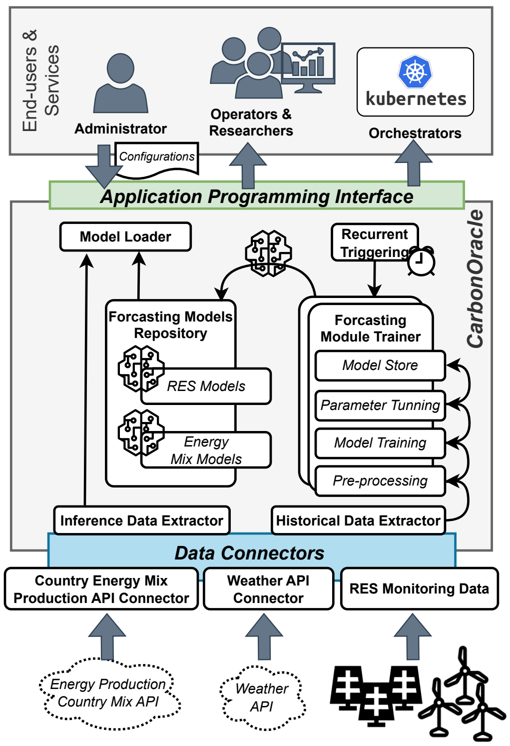

Fig. 1. CarbonOracle Service Overview

asynchronously starts the extraction of the historical data, like weather, energy production, and energy mix data, from online sources and self-hosted monitoring datasets, which will later be injected into the forecasting training pipelines. To do that, the system utilizes the *Historical Data Extractor*, which is responsible for combining the data from various sources into a single dataset. Specifically, the *Historical Data Extractor* invokes a set of *Data Connectors*, each of which materializes the data connector interface and retrieves data from a distinct source. Currently, the system implements three data connectors, namely, *Country Energy Mix Connector*, *Weather Data Connector*, and *RES Monitoring Data Connector*. For *Country Energy Mix Connector* and *Weather Data Connector*, we utilize website scraping and online APIs, while RES data is retrieved from a self-hosted monitoring server in which we export PV metrics data (see Sec. IV-B). Having the latter data on hand, the service combines them into datasets, one for each RES and one for each country's energy mix.

Next, datasets are injected into the *Forecasting Training Modules*. Users can design different Forecasting Training Modules, with the system currently providing by default two modules, one for PV RES and one for country energy mix training. Each module performs a set of automated steps to create each forecasting model. Specifically, the first step is the pre-processing of the data, including the removal of duplicate values, handling of missing values, encoding of features, and so on. Then, the preprocessed datasets are passed through the training step, in which CarbonOracle evaluates a set of ML/AI models and selects the best of them. The third and fourth steps include the (optional) hyper-parameter tuning of the best model, generating the optimal model for each case, and the model storing, respectively. CarbonOracle stores and updates the forecasting models in its *Forecasting Models Repository*, having them readily available for inference.

After the training, CarbonOracle can "answer" the questions about the energy production and carbon emissions of the pre-defined RES and energy grids. Specifically, CarbonOracle exposes a RESTful API allowing users and other systems, like dashboards or Cloud schedulers (e.g., Kubernetes), to have access to it and its predictions. An external system can perform an API query providing parameters, like the RES-id or the specific country, along with the forecasting period in hours. Having these parameters, CarbonOracle invokes the *Model Loader*, which loads the respective forecasting model from the Forecasting Models Repository and retrieves all needed data from the online sources. A separate component is used for that, namely the *Inference Data Extractor*. This component utilizes again the implemented *Data Connectors* to retrieve the respective data for the specific time period, e.g., weather forecasts and/or energy mix historical data. Then, *Model Loader* introduces the retrieved data to the respective model and obtains the model's predictions. If the data are related to RES sources, CarbonOracle returns the response as the answer of the CarbonOracle's API. If forecasts are for the country's energy grid, CarbonOracle not only provides the results but also calculates the current greenhouse gas emissions factor, providing a more comprehensive answer to the end user.

IV. Implementation Details

In this section, we describe a set of crucial implementation details about CarbonOracle, namely: (i) the Service's *Application Programming Interface*; (ii) the *Integration of Data Sources*; (iii) the *Creation of the Datasets*; (iv) the *Forecasting Methodologies & Evaluation Metrics*; and (v) the *Model Serving, Inference, and Carbon Intensity* process.

A. Application Programming Interface (API)

Table I depicts the main API endpoints of the CarbonOracle service that are designed for managing, retrieving, and updating renewable energy forecasting and carbon emissions models. The first endpoint that one should use is `/api/config`, which supports both POST and PUT methods, allowing developers to initialize or update the service's configuration. Through this endpoint, key service parameters, such as API keys for an external data source, can be set or modified to ensure proper operation. Another endpoint related to the system configuration is `/api/res`, which can be used with the POST method. Through it, users can introduce diverse RESs and their configurations. For instance, for new PV panels, users should give the geo-location, the size of the PVs, the PV panel identifier, and so on. Additionally, the `/api/res/{id}` endpoint, using the PUT method, allows for updating the configuration of a particular RES identified by its ID, offering granular control over individual RES resources. To retrieve data, the API offers multiple GET endpoints, like the `/api/res` endpoint, which allows users to fetch a list of available RES, providing an overview of

HTTP Method	Endpoint	Parameters	Output	Description
`POST` & `PUT`	`/api/config`	The initial configuration of the service.	-	Sets or updates the service's configuration, such as API keys for external data sources.
`POST`	`/api/res/`	The initial configurations of RES, such as location, characteristics, etc.	-	Introduces the configurations for data extraction and scraping process for RES data.
`PUT`	`/api/res/{id}/`	Renewable source ID as parameter.	-	Updates a specific resource's configuration or forecasting settings identified by ID.
`GET`	`/api/res/`	-	RES data and forecasting models.	Retrieves a list of available RES data sources and forecast models.
`GET`	`/api/res/{id}/`	Renewable source ID as parameter.	Specific RES data or forecast model.	Retrieves detailed data or forecasts for a specific resource identified by ID.
`GET`	`/api/mix/{grid}/`	Energy grid identifier as parameter.	Energy mix forecast data.	Retrieves the energy mix (e.g., PV, wind, total) forecast data for a specific energy grid.

TABLE I
CARBONORACLE API ENDPOINTS

them and their configurations. For more specific details, the `/api/res/{id}` endpoint can be used to retrieve forecasts for a particular resource identified by its ID and a particular time period if the user specifies the forecast horizon. Lastly, the `/api/mix/{grid}` endpoint returns the energy mix forecast for a specific energy grid, including details about energy production such as solar (PV) and wind energy production, and hourly carbon intensity, while again users can introduce a specific forecast horizon.

B. Integration of Data Sources

Next, we describe the integration of our prototype with different sources. To integrate an external data source into CarbonOracle, one must create a class that implements the *DataConnector* interface and its *get_data* method, which returns the data in a timestamped tabular format. Currently, we created three connectors for (i) online historical and forecast weather data; (ii) PV data from three PV self-hosted racks of panels; and (iii) historical country energy mix data. Since our PV panels are hosted in Cyprus, we also focus on the respective regional energy grid for energy mix data.

Online historical and forecast weather data. There are plenty of online APIs that provide both historical and forecasted weather data. Without harming the generality of our solution, we implement the *Weather Data Connector* by adopting weather data from the open-meteo API[1]. On each request, the connector retrieves data like solar irradiance parameters, wind speed, the solar zenith angle, surface orientation, air temperature, etc., for a specific geo-location.

RES data from PV panels. We use data from a real-world deployment of three racks of PV panels installed on the rooftop of our micro-DC, located in the capital of the Republic of Cyprus. The PV panel racks have different sizes, specifically, rack-1 (PV1) is 107.1 m^2, rack-2 (PV2) is 95.2 m^2, and rack-3 (PV3) is 57.8 m^2. The power production data goes through a Fronius inverter[2] that exposes the respective metrics via its HTTP API interface. To capture in real time the panels' data, we build a service that periodically pings the Fronius API and stores the retrieved data in a time-series monitoring server, namely Prometheus[3]. Having stored this data, CarbonOracle's PV data connector communicates with Prometheus API, which allows the retrieval of the data for specific periods.

[1] https://open-meteo.com/
[2] https://www.fronius.com/
[3] https://prometheus.io/

Energy Mix Data. In order to retrieve the energy mix data from the energy grid, this connector gathers (scrapes) its data from the publicly available website of the Cyprus Transmission System Operator [18]. Specifically, CarbonOracle can give a time range, and the connector retrieves the hourly total energy consumption and energy production from wind and PVs for this period. Unfortunately, the website allows us only to see data for a specific date, so the connector downloads, one by one, the HTML pages with the respective daily measurements until it gathers all related data and then transforms the data into a tabular form.

C. Creation of the Datasets

Since the energy generation from RES sources and the energy mix of the country's power grid are weather-dependent, we combine weather data with energy mix and RES data to create the required datasets for the forecasting models' training. Due to the nature of these datasets, we follow two approaches, creating two different categories of datasets, one for each *Forecasting Trainer Module*. Our rationale is based on the fact that a RES unit (like a PV panel) is placed in a specific location, so the corresponding environmental conditions, like localized weather data, should be tailored to that particular site. In contrast, modeling the energy mix for an entire country requires weather data from multiple locations, especially from the sites of major PV or wind parks.

Starting from the RES forecasting models (PV panels in our case), CarbonOracle retrieves weather data for the geolocation of each RES over a selected time period. We should note here that both geolocation and assigned period should be configured by the user at the initialization of the service. Then, the service creates a dataset by joining them on each measurement timestamp. At the current state of the CarbonOracle service, the measurements are captured on an hourly base, so the aforementioned timestamps represent a specific hourly period, and our predictions have the same granularity. The final RES-weather dataset includes a column with the RES energy production measurements and a set of 27 columns of weather data. Moreover, we include other relevant parameters, such as the time of sunrise and sunset, the hour of the day, the month of the year, etc. Afterward, the system creates a separate dataset for each RES. In our example, this includes three PV racks, so the system generates three individual datasets.

For the energy mix, CarbonOracle utilizes a slightly different approach. Given that the energy production from renewable sources is currently dominated by large-scale RES parks (e.g., PV or wind turbine parks) and energy consumption is concentrated in large cities, which host the majority of a country's population, users should define the geolocations of these parks and the largest cities during the initialization of the CarbonOracle service. We configured these settings to include the locations of Cyprus's three largest PV and wind parks for energy production, along with the three most populated cities for energy consumption. The system then retrieves specific weather data for these locations (e.g., it will retrieve only wind-related data for the wind parks) along with the data for the country's electricity grid. Finally, CarbonOracle combines the production data and grid consumption data with the weather data of the parks' and the cities' locations, respectively, creating three (PV, wind, and total consumption) datasets. Again, energy mix datasets provide hourly granularity, aligning with the resolution of RES datasets to ensure consistency in CarbonOracle's forecasting models.

D. Forecasting Methodologies & Evaluation Metrics

To create our models, we consider two distinct ML methodologies, one for RES and one for the grid energy data, because the behavior of a single RES is changing more slowly than the changes in a whole country. For that reason, the modeling part of the energy mix should take into account the temporal characteristics of the time-series data, which is not that crucial for the RES models. Next, we provide technical details about the materialization of the respective training methods and the AI/ML forecast evaluation metrics.

Renewable Sources (PV panels). In the case of RES, we use regression modeling with the target metric being the energy generation of each RES and features being the forecasted weather data extracted from the aforementioned weather API. Specifically, we use an autoML library, namely PyCaret [19]. This library automatically optimizes a set of selected AI/ML models performing hyperparameter tuning techniques, k-fold cross-validation, and extracting predefined or user-created ML performance metrics, such as Mean Absolute Error (MAE), Mean Average Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), etc.

Grid Energy Mix and Consumption. For the energy mix, we selected our models to take as input not only weather forecasts but also historical data, so we focus more on time-series models. Specifically, we utilize a library specialized for time-series models, namely skforecast[4]. This library allows the use of diverse ML models, including popular options like LightGBM, XGBoost, and CatBoost, among others. Despite the wide range of skforecast features, our analysis focuses on the ability to build models based on historical values (lags) and the ability to provide exogenous features to the forecasting models. On the one hand, historical values allow a model to adapt its output based on recent target metric values. For example, we set lags to be 24 values in all models considering the last-day metrics. According to our research, a larger historical horizon does not introduce any important improvement to the models. On the other hand, exogenous variables are outputs of independent predictors that are not part of the forecasting model itself. To incorporate them into the prediction process, their future values must be known in advance. In our models, the exogenous variables come from online weather data forecasts. According to the models that we evaluated, the best performance seems to be provided by tree-based ensembles, namely Random Forest, Gradient Boosting, and Extreme Gradient Boosting, compared with simpler models like a linear regression model or a simple decision tree. For that reason, our analysis in Section V focuses on these models, while also the system automatically performs a hyper-parameter tuning on the model with the best results.

Forecast Evaluation Metrics. In order for users to know the performance of the generated models, CarbonOracle provides the following metrics in the model selection process: (i) *Mean Absolute Error (MAE)*, measures the average magnitude of the errors between predicted and observed values, offering an easy interpretation of the accuracy by highlighting errors in the same units as the data; (ii) *Root Mean Square Error (RMSE)*, penalizes larger errors as it squares the differences before averaging them, this metric is susceptible to outliers; (iii) *R-squared (R^2)*, used as an indicator showing the proportion of the variation in the dependent variable that is predictable from the independent variables; (iv) *Mean Absolute Percentage Error (MAPE)*, a statistical measure that expresses the prediction error as a percentage, allowing for easy comparison across different datasets. However, as the MAPE computation relies on division of the actual value, in cases where this value is 0 (i.e., no PV production) results can be skewed towards values near or equal to 0. To avoid this, we opt to use a *Tuned MAPE – tMAPE* that excludes error computations when the actual value is 0; and (v) *Tuned Symmetric Mean Absolute Percentage Error (tSMAPE)* is a percentage-based metric that quantifies forecast accuracy by comparing absolute differences between predicted and observed values, normalized by their average, ensuring balance even when actual values are near zero. Users can select the most important to them, and the system optimizes the forecasting models accordingly. We set *MAE* as the default parameter tuning metric.

Calculation of the Evaluation Metrics. For PV panels, the system utilizes k-fold cross-validation on the respective datasets to evaluate regression models. On the contrary, the forecasting of the grid energy mix takes into account not only the weather data but also historical data of the predicted values. So, one cannot simply use the same metrics but needs to perform a realistic retrospective evaluation on historical data, or in other words, a backtesting method. This method needs the prediction horizon (how many values in the future our model should be able to predict), and the refitting strategy (how often we will retrain our model). For the evaluation of these models, we consider a scenario where we would like to predict the hourly energy mix for the next three days with a step of one hour. In this process, users can select if the refitting of the

[4] https://skforecast.org

models would be enabled, so the system will emulate a more realistic evaluation with retraining for specific periods but sacrificing more time for training, or if refitting is disabled, the model is trained once (faster), and validation is performed on a subset of data. By default, CarbonOracle does not use refitting in its models because we did not observe any significant performance improvement when we tested this approach.

E. Model Serving, Inference, and Carbon Intensity

With the models generated and stored, the system is able to retrieve the respective model and, based on the user's preferences, to forecast its predictions. For the RES models, the system just fetches the model in memory, requests the weather API by utilizing the Weather API Connector, and the model generates the values based solely on the retrieved weather predictions. For the country energy mix, CarbonOracle follows a similar approach by fetching the respective model in memory and retrieving forecast weather data but also needs historical data on energy mix and consumption. So, CarbonOracle utilizes the Energy Mix Connector to gather that data, with the Energy Mix Connector retrieving (scraping) the requested values. It should be noted that our methodology currently focuses on predicting PV and wind from the electricity grid, because in Cyprus the rest of RES, like biomass, are negligible, and we set a static energy production value, e.g., the average produced energy per hour for the last 2 years.

For the value of carbon emission measurements, CarbonOracle multiplies the generated energy from different sources with a set of source-based emission factors. An emission factor is a coefficient that describes the rate at which a given activity releases greenhouse gases into the atmosphere. It is the average emission rate of a given source relative to units of activity or process/processes. Even renewable sources introduce emission factors since their calculation includes upstream processes (e.g., the extraction of raw materials for RES creation), operational processes (e.g., system operation and maintenance), and downstream processes (e.g., RES park decommissioning) [20]. So, after the forecast of grid energy mix predictions, CarbonOracle computes the carbon intensity for one kWh by translating the generated values into the percentage of the total energy consumption and combines them with emission factors based on data reported in [21].

V. Evaluation

In this section, we focus on the evaluation of our methodology and its performance. To evaluate the performance of the RES and energy grid forecasting methodologies, we configure CarbonOracle for our RES-enabled micro-DC. Specifically, our micro-DC is placed in Cyprus and includes three racks of PV panels. So, we configure the CarbonOracle service to create forecast ML models from our PV racks and from Cyprus's energy mix (see Sec. IV). Then, we configure the system to log its evaluation results for all models.

A. RES Forecasting Methodology Performance

For evaluating the performance of RES regression models, CarbonOracle captured metrics every hour from the three racks of PV panels for 135 days (3240 data points). Also, we manually extract an evaluation dataset that includes observations of 7 days (26/06/2024-02/07/2024) and utilize it only to show the performance of the final (best) models on unknown data. Next, we focus on finding already created models for PV energy production modeling in order to utilize them as a baseline. We found the PVWatts formula [14], [15] (`PV Formula`) that estimates the energy output of a PV, based on environmental parameters and several parameters related to the PV system's configuration and performance. Moreover, this formula aggregates multiple factors that reduce the system's efficiency, such as inverter losses, wiring losses, soiling, shading, and the effects of sun-tracking and aging. Since we are not aware of all these system-specific parameters, we selected to perform a hyper-parameter tuning method (using Optuna library[5]) to find an optimal set of parameters. Moreover, we evaluated a simpler model proposed in [22], which only takes into account the current irradiation multiplied by the area of PV panels and their efficiency. To calculate our panels' efficiency, we divide the maximum energy production for each PV panel with the maximum irritation in the dataset. It is important to note that CarbonOracle forecasts, as well as `PV Formula` and `Naive Model`, are not limited by the forecast horizon. The only requirement is having accurate weather predictions for the forecast period. To this end, Table II shows the metrics of top-6 models for each PV panel along with the results of the `PV Formula` and `Naive Model`.

Model	MAE	RMSE	R^2	tMAPE	tSMAPE
PV1					
Random Forest	392	953.71	0.929	**0.37**	**0.073**
Decision Tree	506	1248.16	0.878	0.372	0.09
Extra Trees	**385**	**939.27**	**0.931**	0.567	0.097
K Neighbors	440	1038.83	0.916	0.524	0.12
XGBoost	405	974.88	0.926	1.317	0.588
Gradient Boosting	417	965.71	0.927	2.904	0.592
Baselines					
Naive Model	941	1832.55	0.764	0.841	0.1645
PV Formula	458	1021.89	0.926	0.247	0.129
PV2					
Random Forest	365	891.51	0.923	0.322	**0.091**
Decision Tree	477	1173.03	0.866	**0.315**	0.104
Extra Trees	**359**	**879.60**	0.924	0.368	0.114
K Neighbors	392	947.41	0.913	1.110	0.1307
Gradient Boosting	379	909.10	0.918	2.459	0.5924
XGBoost	380	938.20	0.915	1.624	0.5967
Baselines					
Naive Model	914	1763.70	0.730	0.972	0.1737
PV Formula	406	908.46	**0.928**	0.268	0.1343
PV3					
Random Forest	212	518.74	0.924	**0.621**	**0.0898**
Decision Tree	276	673.50	0.872	0.632	0.1028
Extra Trees	**205**	**504.65**	**0.928**	0.737	0.1114
K Neighbors	233	560.28	0.912	0.852	0.1267
XGBoost	223	552.12	0.915	2.491	0.6060
Gradient Boosting	224	519.69	0.924	4.344	0.6107
Baselines					
Naive Model	597	1197.55	0.634	3.507	0.1796
PV Formula	300	646.68	0.893	0.716	0.1427

TABLE II
Performance Metrics of PV RES Regression Models

Interestingly, for all panels, the best model according to

[5] https://optuna.org/

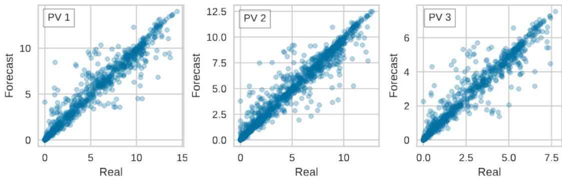

Fig. 2. Real VS Forecast Values of PV panels

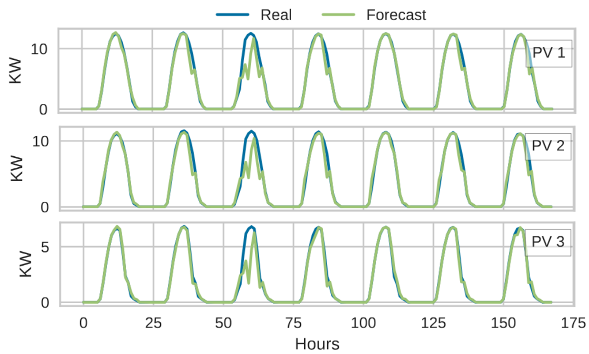

Fig. 3. Real VS Forecast Values of PV panels for 7 days

tSMAPE is Random Forest with 7.3%, 9%, and 8.9% for PV1, PV2, and PV3, respectively. Moreover, more than half of the evaluated ML/AI models outperform both `Naive Model` and `PV Formula`. Specifically, `Naive Model` provides a tSMAPE about 16-18% for all PVs and `PV Formula` ranging from 12.9% to 14.2%. We observe a similar pattern for the rest of the metrics, with most of the AI/ML models outperforming both `Naive Model` and `PV Formula`, except for R^2 of `PV Formula` results for PV2 panel.

According to tMAPE, Random Forest is almost always better with an exception being PV2 data, on which Decision Tree has better tMAPE. For the rest of the metrics, the Extra Trees regressor has the best performance in all cases. Specifically, the Extra Trees regressor has a 0.93 R^2, followed by Random Forest and Gradient Boosting with 0.929 and 0.926, respectively. Another important metric for PVs is MAE which indicates the absolute difference between forecast and real values. Again in this metric, Extra Trees provide the best results (PV1=385, PV2=359, PV3=205), followed by other tree-based models, like Random Forest (392, 365, 212), XGBoost (405, 379, 223), and Gradient Boosting (417, 379, 224).

Figure 2 presents scatter plots comparing 2K randomly selected real points against the forecast values of the best model based on MAE (Extra Trees Regressor) across our three PV panels. Each subplot displays the most of points being around the diagonal line, indicating a strong correlation between the real and forecast values, with a small spreading of points highlighting a good model fitting. Furthermore, Figure 3 depicts the predicted and actual energy production values of PVs for 7 days (evaluation dataset). The close alignment between the blue (real values) and green (forecasts) curves across all three PV panels suggests that the models accurately capture the patterns in the power output. On a closer examination of the third day, the forecast values show a noticeable drop in power output, but the real values do not follow the same trend. The latter may happen due to inaccurate

Model	MAE	RMSE	R²	tMAPE	tSMAPE
Wind					
Random Forest	11.30	16.16	0.49	0.9740	0.3137
Gradient Boosting	11.03	16.02	0.5042	0.9299	0.3108
XGBoost	11.81	17.32	0.42	0.8795	0.3278
Tuned Model (GB)	**10.72**	**15.63**	**0.52**	**0.8739**	**0.3011**
PV					
Random Forest	22.35	42.37	**0.90**	**0.1383**	**0.0808**
Gradient Boosting	22.46	42.45	**0.90**	0.1428	0.0843
XGBoost	24.02	45.28	0.88	0.1484	0.089
Tuned Model (RF)	**21.97**	**40.89**	**0.90**	0.1420	0.0854
Total					
Random Forest	28.13	42.46	0.86	0.0488	0.0250
Gradient Boosting	24.06	35.18	0.91	0.0426	0.0216
XGBoost	26.21	38.28	0.89	0.0468	0.0238
Tuned Model (GB)	**22.53**	**32.85**	**0.92**	**0.0404**	**0.0203**

TABLE III
WIND, PV, AND TOTAL FORECAST METRICS

weather forecasts or local weather conditions, indicating a potential limitation in our methodology, which totally relies on highly accurate weather observations.

Key Takeaway: Through our evaluation, we highlighted that *our methodology outperforms current approaches and simple models that are used for workload management in geo-distributed data centers providing much better performance metrics and reducing the mean absolute error by at least 12%.*

B. Grid Energy Mix Forecasting Methodology Performance

For the energy mix, CarbonOracle is configured to initially gather data between 01/01/2022 and 30/5/2024 that includes hourly total energy consumption and energy production from wind and PVs. Then, the system divides the dataset into three sub-datasets, one for each dimension (total, wind, PV), and applies the energy grid forecasting methodology. Following a feature selection process, CarbonOracle keeps only the relevant weather variables for each dataset, namely: irradiation-related data for PV production, wind speed data for wind farm production, and temperature/humidity data for the total consumption. However, contrary to the PV panels modeling, there are no available forecasting models for the Republic of Cyprus grid energy mix. So, we only investigate different models of our methodology on historical real-world data. To calculate the evaluation metrics, the system generates a prediction for a 72-hour horizon, as described in Sec. IV-D.

Table III introduces the results of examined models for Wind and PV energy production along with the Total energy consumption. Starting from the wind-generated energy, we observe that Gradient Boosting outperforms Random Forest and Extreme Gradient Boosting (XGBoost) in MAE, RMSE, and R^2, with 11.03, 16.0, and 0.50, respectively, compared to 11.3/16.16/0.49 of Random Forest and 11.81/17.32/0.42 of XGBoost. Unfortunately, the low values of R^2 indicate a poor fit of all models which is also depicted in high tSMAPE error values (0.31, 0.31, 0.32). Next, the system tried to improve the best model according to MAE by performing hyper-parameter tuning on the Gradient Boosting model achieving better results, like 10.72 MAE, 0.52 R^2, etc. Given the limited role of wind energy in Cyprus's energy mix and the variable nature of wind (e.g., lack of a diurnal pattern and frequent

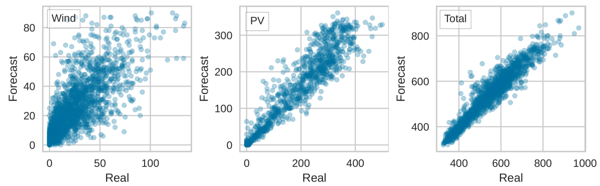

Fig. 4. Real VS Forecast Values of Wind, PV, and Total Energy

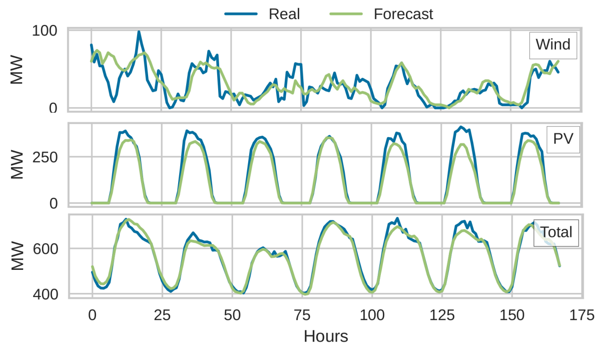

Fig. 5. Real VS Forecast Values of Wind, PV, and Total Energy for 7 days

short-term wind speed fluctuations), CarbonOracle models are bound to have slightly lower accuracy. This effect could be mitigated by using a larger sample size during training.

On the contrary, the auto-generated PV models provide much better results, with the best model based on MAE being Random Forest (22.35), followed by Gradient Boosting (22.46) and XGBoost (24.02). Except for that, Random Forest outperforms the rest of the models in all performance metrics. Thus, CarbonOracle automatically selected Random Forest as the best model for hyper-parameter tuning, generating a tuned Random Forest model. The tuned model provided better MAE (21.97) and RMSE (40.89), and the same R^2 (0.90), but worsened tMAPE (14%) and tSMAPE (8.54%).

Lastly, CarbonOracle generates forecasts for the Total energy consumption of the country. In our case, the system provided the best results, compared to the performance of wind and PV models. Specifically, Gradient Boosting offers the best values in all metrics: 24.06 MAE, 35.18 RMSE, 0.91 R^2, 4.26% tMAPE, and 2.16% tSMAPE. XGBoost model is in second place in all metrics, and the worst performance is provided by the Random Forest model. Following hyper-parameter tuning, the Gradient Boosting metrics improved, achieving a 22.53 MAE, 0.92 R^2, and 2% tSMAPE.

Figure 4 depicts 2K randomly selected real and predicted values from our dataset for PV and wind energy generation and the grid energy demands. Clearly, the worst results are given from wind energy production. PV energy production provides better results, and total energy consumption forecasts highlight the best correlation between forecasts and real values. Moreover, Figure 5 presents the actual and predicted values for PV and wind energy generation, as well as total consumption, over the course of one week. As we can see, wind-based generation does not provide any periodicity, making it difficult for our model to make the correct predictions. The latter shows a poor performance of wind forecast models. On the contrary, PV energy generation and Total energy consumption models closely follow the same trend as the real data.

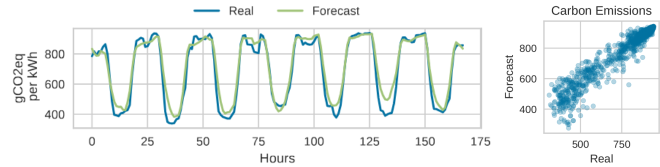

Fig. 6. Real VS Forecast Carbon Emission Values for 7 days

Key Takeaway: To this end, *CarbonOracle seems to provide high accuracy for PV energy production and total energy consumption with an MAE of 21.97 (about 8% tSMAPE error) and 22.53 (about 2% tSMAPE & 4% tMAPE error), respectively. However, when an energy source is limited and unpredictable, like wind-related energy production, the service forecasts present a slightly higher reported error.*

C. Computing Carbon Emissions

As we described earlier, CarbonOracle translates the predicted values from the grid energy mix models into carbon emissions by utilizing the already-known carbon intensity of each source. The carbon emissions are measured in gCO2eq/kWh, which refers to the grams of $CO2$ equivalent emitted per kilowatt-hour of energy produced, providing a standardized measure to compare the environmental impact of various greenhouse gases. This metric is commonly used to assess the carbon footprint of electricity generation sources. In this section, we compute and compare the carbon emissions calculated from real-world data and predictions of our models.

Even if the results of carbon emissions were generated after the sub-models inference, we computed all relevant performance metrics (MAE, RMSE, R^2, tMAPE, tSMAPE). Specifically, tMAPE and tSMAPE highlight a percentage error of 8.2% and 3.7%, respectively. Furthermore, the high R^2 (0.88) indicates that a large proportion of the variance in the dependent variables is explained by the independent variables in the underlying models, reflecting a strong forecast fit. Lastly, MAE and RMSE are 44.59 and 64.78, respectively.

To visually evaluate our method, plots of Figure 6 present a comparison between real and forecasted carbon emissions in gCO2eq per kWh. In the left plot, the real (blue line) and forecasted (green line) values follow a similar pattern, with the model capturing the periodic fluctuations in carbon emissions. Although minor inconsistencies exist, the overall alignment between the two curves indicates a high level of forecast accuracy. The scatter plot on the right depicts 2K randomly selected points, and their distribution further supports the strong positive correlation between real and forecasted emissions, as the data points closely follow the diagonal line.

Key Takeaway: The latter results show that *even if some models for energy production do not provide good results, like the wind-production model, the overall accuracy in the prediction of carbon emissions of the energy grid is high, allowing practitioners and systems to make accurate decisions based on them.*

VI. Related Work

Our work relates to modeling RES & energy grid carbon emissions, as well as carbon-aware schedulers.

A. Energy Grid Carbon Emissions and RES Modeling.

Various efforts have been made to predict carbon emissions and optimize RES energy usage, however often with limited prediction horizons. Web services like Electricity Map [17] and ENTSO-e [16], as well as models by research papers, like [23] and [24], focus on one-day forecasts, which restrict their usefulness for longer-term planning. Similarly, other efforts, like [25] and [26], aim to optimize grid operation and improve carbon emission forecasts by utilizing only day-ahead weather data, enhancing accuracy but still focusing on short prediction windows, while authors in [27] use time-series models excluding weather data. Recently, efforts have applied DL models to improve forecast accuracy and extend prediction horizons. For instance, authors in [28] introduce a Mixture-of-Experts (MoE) model using LSTMs, CNNs, and transformers, achieving significant accuracy improvements in load and PV forecasts, demonstrating the potential of specialized models used for different energy factors. Furthermore, CarbonCast [29] uses a hierarchical ML approach to forecast grid carbon intensity for up to four days. It employs neural networks for production forecasts of electricity sources and a hybrid CNN-LSTM to integrate these with historical carbon data and weather forecasts. In optimizing energy usage, Pahlevan et al. [30] use linear models to predict PV output for DCs, while authors in [31] focus on forecasting weather data impacting RES, though without modeling PV's energy generation.

While significant advances have been made in ML and forecasting techniques for RES and grid carbon emissions, several gaps remain, which our research fulfills, namely: *(i) no existing work combines forecasting models for both RES and grid carbon emissions in a unified system; (ii) there is a lack of automated solutions to streamline the creation of these models, reducing the complexity involved; and (iii) most state-of-the-art methods are constrained by short prediction horizons, typically limited to a single day, which limits their practical applicability for long-term energy planning.*

B. Carbon-aware and Energy-aware Load Schedulers

Several works focus on reducing energy consumption and carbon emissions in computing systems, such as parallel systems [32], ML models [33], [34] and distributed analytics [12], [35]. In [35], an FL carbon-aware scheduler is introduced, aiming to optimize training within a fixed carbon footprint budget. Through simulations using real-world carbon intensity data, the authors optimize learning performance while minimizing environmental impact. However, the work only considers the current carbon intensity without integrating any forecasting algorithms. Focusing again on FL training, the authors of [13] propose an FL client selection strategy that minimizes carbon emissions by leveraging forecasts of computation loads and renewable energy production. Their solution is robust to forecast inaccuracies, a common issue in renewable energy forecasting. For RES predictions, they use data from Solcast[6] that offers limited free API calls and provides generalized predictions that are not tailored to specific PV panels.

[6] https://solcast.com/

Several works have utilized time-series models for energy forecasting. The system proposed in [36] employs existing time-series models to migrate workloads between data centers, using models similar to those in [27]. In contrast, [22] presents an optimization approach for energy-efficient resource allocation in mini DCs, exploiting VMs and energy migrations between green computing nodes. Notably, the authors of [11] present a framework that optimizes carbon efficiency for applications across edge-cloud infrastructures. It primarily focuses on reducing carbon emissions by accounting for the variability of renewable energy sources, location-based carbon intensity, and runtime fluctuations. Although, it does not incorporate forecasting algorithms for renewable energy sources or energy grid carbon emissions focusing only on real-time optimization based on the available data. Wiesner et al. [12] examine shifting computational workloads to times when energy is predicted to be less carbon-intensive. Their findings show that larger forecast windows improve outcomes, but they note a lack of public forecasting tools for grid carbon intensity across regions. To assess inaccurate forecasts, they added artificial noise to the carbon intensity data, revealing current limitations. In [37], the authors propose a self-adaptive resource management approach to reduce data center carbon footprints by minimizing brown energy use and maximizing renewable energy consumption. Their system reduces brown energy usage by 21% and increases renewable energy usage by 10%. However, they use a simple support vector machine (SVM) regressor to predict solar irradiation rather than the actual energy production of a PV panel.

The aforementioned systems reduce carbon emissions for DCs or specific workloads (e.g., FL) relying solely on pre-existing forecasts or simplistic models with limited focus on accuracy. So, despite the advancements in energy-efficient computing and carbon-aware scheduling, the integration of reliable forecasting mechanisms for RES and carbon emissions in energy grids remains an open challenge. To this end, CarbonOracle tackles this challenge and helps carbon-aware systems by offering *an automated ML model creation, delivering more accurate predictions, and extending forecast horizons. This approach aims to enrich both the accuracy and efficiency of decision-making algorithms in energy management.*

VII. Conclusion and Future Work

In this paper, we introduced CarbonOracle, an autoML service designed to support carbon-aware DC management by providing accurate forecasts for renewable energy production and grid energy mix. By automating the extraction of data from self-hosted RES, online energy grids, and weather APIs, CarbonOracle simplifies the process of deploying sustainable and efficient services across geo-distributed micro DCs. Our evaluation shows that CarbonOracle achieves promising accuracy rates for PV panel production forecasts and the country's grid carbon emissions, demonstrating its potential for real-world applications. Additionally, by integrating AI/ML methodologies, CarbonOracle addresses the current limitations in long-term forecasting. The evaluation of our modeling showed an error of about 9% on real-world data from PVs and an error of about 4% on electricity grid carbon emissions.

Our *future work* will focus on expanding our evaluation to include data from multiple regions with diverse climates and energy infrastructures, assessing the system's generalizability. Incorporating new RES such as hydroelectric and biomass will enhance versatility, while exploring advanced forecasting techniques like deep learning models (e.g., LSTMs, GRUs) could improve the prediction accuracy of less reliable metrics, such as wind energy. Moreover, extending forecast horizons and integrating uncertainty quantification will enhance the robustness and utility of the predictions. Finally, CarbonOracle's integration with data center operations, such as container orchestration platforms, will validate practical applicability.

Acknowledgement. This work is part of GreenAnalyzer and AdaptoFlow, which have indirectly received funding from the European Union's Horizon Europe research and innovation action programme, via the aerOS Open Call issued and executed under the aerOS project (Grant Agreement no. 101069732) and the TRIALSNET Open Call issued and executed under the TrialsNet project (Grant Agreement no. 101017141), respectively. Additionally, it is partially funded by the Cyprus Research and Innovation Foundation under the Holo-CIM project (CODEVELOP-ICT-HEALTH/0322/0047).

REFERENCES

[1] N. Jones, "How to stop data centres from gobbling up the world's electricity," *nature*, vol. 561, no. 7722, pp. 163–166, 2018.

[2] IEA. (2024) Energy System: Buildings - Data Centres and Data Transmission Networks. [Online]. Available: https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks

[3] TechRadar Pro. (2024) Generative ai triples the carbon dioxide emissions from data centers. [Online]. Available: https://www.techradar.com/pro/generative-ai-triples-the-carbon-dioxide-emissions-from-data-centers

[4] The Guardian. (2024) Ireland's datacentres overtake electricity use of all urban homes combined. [Online]. Available: https://bit.ly/4e7nwEZ

[5] . (2022) What is the Impact of the War in Ukraine on Europe's Climate and Energy Policy? [Online]. Available: https://tinyurl.com/ysmmm5vz

[6] ——. (2022) UK Net Zero Strategy: Build Back Greener. [Online]. Available: https://tinyurl.com/56mn2847

[7] Google. (2020) Renewable Energy. [Online]. Available: https://www.google.com/about/datacenters/renewable/index.html

[8] Microsoft. (2020) Transitioning to Zero-Carbon Energy. [Online]. Available: https://www.microsoft.com/en-us/environment/energy

[9] Amazon. (2020) Sustainability in the Cloud. [Online]. Available: https://aws.amazon.com/about-aws/sustainability

[10] M. D. Dikaiakos, N. G. Chatzigeorgiou, A. Tryfonos, A. Andreou, N. Loulloudes, G. Pallis, and G. E. Georgiou, "A cyber-physical management system for medium-scale solar-powered data centers," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 10, 2023.

[11] Y. G. Kim, U. Gupta, A. McCrabb, Y. Son, V. Bertacco, D. Brooks, and C.-J. Wu, "Greenscale: Carbon-aware systems for edge computing," 2023. [Online]. Available: https://arxiv.org/abs/2304.00404

[12] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's wait awhile: how temporal workload shifting can reduce carbon emissions in the cloud," in *Proceedings of the 22nd International Middleware Conference*, ser. Middleware '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 260–272.

[13] P. Wiesner, R. Khalili, D. Grinwald, P. Agrawal, L. Thamsen, and O. Kao, "Fedzero: Leveraging renewable excess energy in federated learning," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. e-Energy '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 373–385.

[14] A. P. Dobos, "Pvwatts version 5 manual," 9 2014. [Online]. Available: https://www.osti.gov/biblio/1158421

[15] National Renewable Energy Laboratory, "Pvwatts calculator," 2023. [Online]. Available: https://pvwatts.nrel.gov/

[16] ENTSO-e, "European Network of Transmission System Operators for Electricity Platform," 2024. [Online]. Available: https://www.entsoe.eu/

[17] Electricity Maps, "Electricity Maps Platform," 2024. [Online]. Available: https://app.electricitymaps.com

[18] (2024) Cyprus transmission system operator website. Cyprus Transmission System Operator. [Online]. Available: https://tsoc.org.cy/

[19] M. Ali, *PyCaret: An open source, low-code machine learning library in Python*, April 2020, pyCaret version 1.0.0. [Online]. Available: https://www.pycaret.org

[20] L. C. Stages *et al.*, "Life cycle greenhouse gas emissions from solar photovoltaics," *J. Ind. Ecol*, vol. 16, p. S122, 2012.

[21] Electricity Maps, "Methodology - electricity maps," https://www.electricitymaps.com/methodology#carbon-intensity-and-emission-factors, 2024.

[22] M. D. M. da Silva, A. Gamatié, G. Sassatelli, M. Poss, and M. Robert, "Optimization of data and energy migrations in mini data centers for carbon-neutral computing," *IEEE Transactions on Sustainable Computing*, vol. 8, no. 1, pp. 68–81, 2023.

[23] V. Aryai and M. Goldsworthy, "Day ahead carbon emission forecasting of the regional national electricity market using machine learning methods," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106314, 2023.

[24] K. Leerbeck, P. Bacher, R. G. Junker, G. Goranović, O. Corradi, R. Ebrahimy, A. Tveit, and H. Madsen, "Short-term forecasting of co2 emission intensity in power grids by machine learning," *Applied Energy*, vol. 277, p. 115527, 2020.

[25] J. Gorka and L. Roald, "Classification models for forecasting and real-time identification of solar curtailment in the california grid," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. e-Energy '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 158–169.

[26] D. Maji, R. K. Sitaraman, and P. Shenoy, "Dacf: day-ahead carbon intensity forecasting of power grids using machine learning," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ser. e-Energy '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 188–192.

[27] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks," in *IEEE MASS 2012*, 2012, pp. 75–83.

[28] J. Sievers, T. Blank, and F. Simon, "Advancing accuracy in energy forecasting using mixture-of-experts and federated learning," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. e-Energy '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 65–83.

[29] D. Maji, P. Shenoy, and R. K. Sitaraman, "Multi-day forecasting of electric grid carbon intensity using machine learning," *SIGENERGY Energy Inform. Rev.*, vol. 3, no. 2, p. 19–33, jun 2023.

[30] A. Pahlevan, M. Zapater, A. K. Coskun, and D. Atienza, "Ecogreen: Electricity cost optimization for green datacenters in emerging power markets," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 2, pp. 289–305, 2021.

[31] L. Caroprese, M. Pierantozzi, C. Lops, and S. Montelpare, "Dl2f: A deep learning model for the local forecasting of renewable sources," *Computers & Industrial Engineering*, vol. 187, p. 109785, 2024.

[32] G. Xie, X. Xiao, H. Peng, R. Li, and K. Li, "A survey of low-energy parallel scheduling algorithms," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 1, pp. 27–46, 2022.

[33] D. Trihinas, L. Thamsen, J. Beilharz, and M. Symeonides, "Towards energy consumption and carbon footprint testing for ai-driven iot services," in *IC2E*, 2022, pp. 29–35.

[34] D. Trihinas, P. Michael, and M. Symeonides, "Towards low-cost and energy-aware inference for edgeai services via model swapping," in *2024 IEEE International Conference on Cloud Engineering (IC2E)*, 2024.

[35] J. Bian, L. Wang, S. Ren, and J. Xu, "Cafe: Carbon-aware federated learning in geographically distributed data centers," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. e-Energy '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 347–360.

[36] B. Aksanli, J. Venkatesh, I. Monga, and T. Simunic, "Renewable energy prediction for improved utilization and efficiency in datacenters and backbone networks," in *Computational Sustainability*, 2016.

[37] M. Xu, A. Toosi, and R. Buyya, "A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 04, pp. 544–558, oct 2021.