university of groningen

faculty of science and engineering

Exploring the effects of Location and Time on the Calculation of the Carbon Footprint of Computing

Bachelor's Project Computing Science

July 2024

Author: Konstantinos Chasiotis
Student Number: S4640209
First supervisor: Prof. Vasilios Andrikopoulos
Second supervisor: Dr. Brian Setz

CONTENTS

I Introduction & Motivation 2

II Background 2
- II-A Electricity Generation Dynamics . . . 3
- II-B Emission Scopes & Carbon Emission Factors . . . 3
- II-C Calculation of Carbon Intensity . . . 3
- II-D Impact and Strategies for Emission Reduction . . . 3
- II-E Energy Consumption and Carbon Emission Estimation Tools . . . 4
- II-F Predictive Modeling and Tools for Carbon Emission Estimation . . . 5

III Methodology 5
- III-A Carbon Footprint Estimator Tool Design . . . 6
  - III-A1 Overview . . . 6
  - III-A2 System Architecture & Design . . . 6
- III-B Data Cleaning . . . 8
- III-C Estimation of Carbon Emissions . . . 9
  - III-C1 SPEC Power Benchmark . . . 9
  - III-C2 Calculating Server Emissions . . . 9
- III-D Running and Validating the CarbonCast Forecasting Model . . . 10
  - III-D1 Weather Data Acquisition and Processing . . . 10
  - III-D2 Configuration and Model Execution . . . 11

IV Analysis 11
- IV-A Influence of Energy Mix and Time on Carbon Emissions . . . 11
  - IV-A1 Statistical Analysis of Regional Carbon Intensity Differences . . . 13
  - IV-A2 Relationship between Carbon Intensity & Carbon Emissions . . . 14
- IV-B Analysis of Forecasting Model Performance . . . 14

V Discussion and Future Work 15

VI Conclusion 16

References 16

VII Appendix 18

***Abstract*—In an era marked by escalating climate change impacts, this study addresses and evaluates a novel tool designed to estimate and forecast the carbon emissions of server-based computational tasks. By leveraging real-time and historical data on Carbon Intensity (CI), and integrating it with Power Usage Effectiveness (PUE) and server hardware configurations, the tool provides a platform to assess and manage the environmental impact of data center operations. The system utilizes regional energy mix data across several countries to render detailed insights into the temporal and spatial variability of emissions. Our results demonstrate the significant role of regional energy sourcing and CI in shaping the carbon footprint of computational tasks. This study underscores the potential of targeted, data-driven strategies to reduce the carbon emissions of the ICT sector and highlights the importance of temporal and geographic factors in environmental impact mitigation.**

***Index Terms*—Carbon Intensity, Power Usage Effectiveness, Data Centers, Environmental Impact, Carbon Emissions Forecasting**

I. Introduction & Motivation

Amidst the backdrop of an ever-evolving climate, where temperatures fluctuate and weather patterns shift unpredictably, the natural balance of our environment faces unprecedented challenges. Human activities, from deforestation to industrialization and the burning of fossil fuels, have significantly altered the natural landscape, disrupting ecosystems, threatening biodiversity, and damaging the world economy [1]. These transformations underscore the relationship between climate change and environmental degradation, highlighting the urgent need for comprehensive solutions to mitigate the impacts on our planet's fragile ecosystems.

Since the Industrial Revolution, the widespread use of coal and fossil fuels marked a pivotal moment in human history. Yet, as we have now transitioned to the Fourth Industrial Revolution [2], coal and fossil fuels are not the only resources whose use would result in increased greenhouse gases (GHG) and as a result, global warming. An often underestimated human activity regarding its environmental impact pertains to computing tasks and the operation of data centers. Considering the advancements made in Information and Communication Technology (ICT) in the last 20 years, the increase in consumer devices, networking technologies, and data centers has led to a substantial environmental footprint, contributing up to 3.9% of global emissions [3] [4].

Examples of computational advancements very well include the adoption of cryptocurrencies but also the use of Large Language Models (LLMs). Regardless of the benefits such advancements bring to society, their environmental impact should be highlighted. When it comes to cryptocurrencies like Bitcoin, mining farms have been a major contributor to carbon emissions. With an estimated energy usage of 173.42 TWh between 2020-2021, evaluating to 85.89 Mt of CO_2, this value is almost equivalent to burning 84 billion pounds of coal [5]. This makes Bitcoin's environmental footprint exceed that of most nations [6]. Similarly to cryptocurrencies, LLMs also have their share of emissions, with models such as those of "OpenAI's GPT-3 and Meta's OPT were estimated to emit more than 500 and 75 metric tons of carbon dioxide" [7].

In today's age, a significant portion of software applications and data processing tasks are hosted on servers, a practice that contributes to increased carbon emissions. Recognizing the challenges inherent in influencing embodied emissions—those tied to the lifecycle of hardware production, transportation, and disposal [3]—our focus shifts towards the operational emissions that users can more directly impact. Operational emissions stem from the energy consumption of hardware during computational tasks [3]. To effectively address the urgent concern of reducing these carbon emissions, we propose the development of a tool that provides essential insights into the environmental footprint of such activities.

In this work, we propose a tool that quantifies and forecasts the environmental impact of computational tasks hosted on servers. This tool not only evaluates operational emissions but also showcases the influence of location and time on these emissions, thereby enabling strategic decision-making for scheduling computational activities. By integrating metrics such as Carbon Intensity (CI) and Power Usage Effectiveness (PUE), the tool estimates the operational emissions based on both the hardware configuration of the server and the CI specific to its geographic location at any point in time, even for future timeframes. Our solution incorporates data on energy mix across multiple regions—including the Netherlands, Germany, Finland, Sweden, Spain, and Poland. This geographic diversity allows us to assess the environmental impact of server operations under varying conditions and different climates. The ultimate goal is to enable the strategic planning of computational activities so as to align them with periods of reduced carbon intensities. Through this tool, we aim to catalyze a shift in user behavior and encourage workload shifting as a strategy for reducing carbon emissions from digital activities.

This approach allows users to understand and manage the carbon footprint of their computing tasks, allowing them to align their operational activities with periods of lower carbon intensity. The paper begins with Section II, which provides a background on the theories and literature concerned in dealing with the environmental impact of computational tasks. Section III details the methodology used to develop the estimation tool and the forecasting model. Section IV presents the results obtained from both the estimation tool and the forecasting model, assessing their accuracy and applicability in real-world scenarios. Then, Section V, outlines future work and improvements to enhance the tool's functionality and accuracy, providing a roadmap for subsequent research efforts in this area. Lastly, Section VI, summarizes our findings and concludes the paper.

II. Background

Before beginning our analysis, it is essential to establish an understanding of the theories relevant to our research and review the existing state of the art. This section first introduces concepts including the structure of electricity grids, the various sources of electricity, and emission factors—all of which

Source	Coal	Oil	Natural Gas	Nuclear	Solar	Wind	Hydro	Other	Biomass	Geothermal
Direct Emission Factor	760	406	370	0	0	0	0	575	0	0

TABLE I: Direct Emission Factors (g/kWh) for Various Energy Sources [8].

contribute to the carbon intensity of electricity. We explore how carbon intensity is calculated and why it varies across different regions and times, reflecting the dynamic nature of energy production and consumption. With this background, we then discuss past literature on carbon emission estimation tools and forecasting models developed for such use.

A. Electricity Generation Dynamics

The most important variable of our research is Carbon Intensity. Carbon Intensity (CI) is a dynamic measure representing the amount of carbon emitted per unit of electricity generated (g/kWh), varying significantly with location and time. This variability arises from the nature of electricity production and the energy sources utilized across different regions. Because vast amounts of energy cannot be efficiently stored, electricity must be generated as it is consumed. This mandate makes it necessary for the power grid, which consists of the phases of generation, transmission, and distribution to be built with the ability to dynamically adapt to changes in demand [9]. Such flexibility is essential, as it allows the grid to adjust to daily and seasonal energy use changes. Considering that regions can generate electricity via sources of the following groups: coal, oil, natural gas, nuclear, solar, wind, hydro, biomass, geothermal, and others, the strategic use of these sources is affected by their availability and the specific energy demands at any given time [10]. This diversity allows the grid to leverage different technologies and resources to meet the energy demands efficiently [9].

The mix of these sources and the efficiency of the grid play pivotal roles in shaping the carbon intensity of the energy supplied. As we consider the environmental implications of different energy mixes, it becomes essential to explore the categorization of emissions associated with electricity generation.

B. Emission Scopes & Carbon Emission Factors

According to the Greenhouse Gas Protocol [11], carbon emissions are split into three distinct scopes:

- **Scope 1**: Includes direct emissions from owned or controlled sources. Emissions of this scope are assumed to be sufficiently small and as a result, they can be safely ignored for our analysis.
- **Scope 2**: Captures indirect emissions from the generation of purchased electricity. So any emissions caused by the production of energy which is purchased and used by a company, fall under Scope 2. This is the focus of our study, particularly relevant for servers where electricity consumption significantly impacts operational carbon footprints.
- **Scope 3**: Encompasses all other indirect emissions in a company's value chain, which, while valuable, the scope they capture is too broad for our analysis on electricity use in servers.

Focusing on Scope 2 emissions allows us to utilize specific emission factors (*EF*, *g/kWh*) for each source (*E*), that quantify the carbon emitted per unit of energy generated by a source. *EFs*, can vary per region and even by power plant [12]. Therefore the calculation of such values may be tedious and difficult to take place accurately. For that reason, defined default emission factors are employed, offering us reasonable estimates [8]. In certain applications like the one presented in this paper, operational (or direct) emission factors are used, which account solely for Scope 2 emissions. The emission factors for direct emissions, as seen in Table I, are crucial for calculating the overall carbon intensity of a region at a particular time.

C. Calculation of Carbon Intensity

Utilizing the emission factors alongside data on the electricity generated by each source allows us to compute the carbon intensity of the electricity produced.

The formula for carbon intensity is given by :

$$\textit{Carbon Intensity} = \frac{\sum(E_i \times EF_i)}{\sum E_i} \tag{1}$$

where E_i represents the electricity generated by source i in megawatts (MW), and EF_i is the carbon emission factor for that source in grams per kilowatt-hour (g/kWh) [12].

Carbon intensity essentially encapsulates the environmental impact of the energy sources used to generate electricity or power. The formula takes into consideration the composition of the energy mix in a given region. Regions relying more on renewable energy sources will demonstrate a lower carbon intensity compared to those dependent on fossil fuels. This aspect of the calculation highlights how the choice and availability of energy sources affect the overall carbon emissions.

Knowing how carbon intensity varies over time and space improves our ability to assess environmental effects and serves as the foundation for energy management strategy decisions. This insight is crucial for infrastructure such as servers, where managing energy consumption to minimize carbon emissions can significantly alter their footprint.

D. Impact and Strategies for Emission Reduction

Time and location have been researched to have a substantial effect on the carbon emissions related to electricity production [13]. To deal with the varying emission levels, theoretical and practical implementations related to transferring workloads of servers to different locations have been researched.

The body of theoretical work seeks to achieve this reduction through the use of Markov decision processes with different loads [14], and linear time-invariant control load systems [15].

Other research has proposed the physical management of data centers and servers to also reduce embodied emissions of such locations, highlighting the need for sustainable use of hardware and software of data centers [16].

Building on these theoretical insights, a practical strategy that has emerged focuses on scheduling computational tasks to coincide with periods of lower carbon intensity within the power grid [17]. This strategy aims to align computing tasks with moments when the power grid is greener. The decision-making process for delaying these tasks considers factors, such as environmental and performance costs. The solution influences how job schedulers decide to prioritize tasks, guiding them on when to line up jobs in a way that supports emission reduction efforts. In other words, it optimizes the server's job scheduler in deciding when and what workloads to queue, as a result delaying jobs' execution [17]. Other methods such as geo-distributed load balancing and data center right-sizing have also been examined [18]. However, these practical strategies are not much of use to us, as they examine sustainable infrastructure practices that could be implemented when building a data center.

E. Energy Consumption and Carbon Emission Estimation Tools

Shifting our focus to methods closely related to our study's goals, it's essential to know how we can accurately estimate the operational emissions from computational tasks. This helps assess the direct environmental impact of our computational activities that run on servers but also assists us in implementing effective energy management strategies. However, direct measurements of emissions from specific computational tasks on various hardware configurations are typically unavailable as raw data. Most of the available data are related to the emissions the server or the data center emits as a whole. The measurements we seek would require access to real-time data on energy use that is often not available or is impractical to collect across the diverse range of hardware setups and operational environments. Due to these limitations and the specificity of each computational task, we try to estimate these emissions. Related to computing such emissions in different settings, one of the most established carbon emission estimators is the GreenAlgorithms tool [19].

Green Algorithms provides a systematic approach to calculating the carbon emissions associated with computing processes, adaptable for both personal computers and servers. When running computations on a personal computer, users must manually input variables such as the core usage factor (u_c) and Power Usage Effectiveness (PUE) [19]. In contrast, computations on servers utilize predefined core usage factors and regional PUE values [19]. To perform the estimation, the tool uses a detailed model that incorporates several components of a computing system's energy consumption:

1. **Computing Cores Power Draw (p_c)**: For the power draw of the computing cores (p_c), the Thermal Design Power (in watts) of the CPU/GPU is used. The TDP is a value provided in the hardware specifications by the manufacturer and indicates the maximum amount of power used by the CPU/GPU when operating at its maximum load [20]. Even though this is not the exact amount of energy that the CPU/GPU will use during the computation of the desired task, it has been shown that it is a good estimate of the energy being used [19].

2. **Memory Power Draw (p_m)**: It was decided to neglect the power draw of the motherboard and the storage as they found it to be minimal. Thus the power draw used is only that of the memory (p_m). The power draw of the main memory has been estimated to be 0.3725 watts per gigabyte and as a result, this constant is used throughout the estimations [19] [21].

3. **Data Center Efficiency (PUE)**: The PUE is a metric that indicates how efficiently a data center uses its power; it is the ratio of the total power used by a data center facility to the power delivered to the computing equipment [22]. A PUE of 1.0 would indicate perfect efficiency, where all power is used directly for computing tasks, while a higher PUE indicates inefficiencies such as power loss in cooling systems, lighting, and other non-computing infrastructure [22]. When calculating the energy consumption of an algorithm, the PUE is multiplied by the energy consumption of the computing resources (cores, memory, etc.) to account for the additional power needed to operate the data center infrastructure. This ensures that the total energy consumption of the algorithm reflects not only the direct energy consumption of the computing resources but also the additional power required to maintain the data center environment.

Other variables used for the calculations include the number of cores used by an individual's CPU (n_c) and the amount of memory the user's system uses (n_m). Additionally, the other variable in the equation is the core usage factor (u_c). This is a metric used to measure the efficiency or utilization of a computer processor's cores in a multi-core system. It quantifies the extent to which the CPU cores are actively processing tasks at a given time. This value is bounded between 0 and 1, and to avoid unnecessary assumptions it is asked to be inputted by the user of the estimator [19]. Calculating the total energy consumed involves combining these elements to reflect the comprehensive energy demand of a computing task:

$$\text{Total Core Power (W)} = n_c \times u_c \times p_c(\text{W}) \times \text{PUE}$$

$$\text{Total Memory Power (W)} = n_m \times P_m(\text{W}) \times \text{PUE}$$

$$\text{Total Power (W)} = \text{Total Core Power} + \text{Total Memory Power}$$

$$\text{Energy Needed (kWh)} = \text{Total Power} \times \text{Runtime} \times 0.001$$

Now to estimate the carbon emissions associated with energy consumption, the energy needed is multiplied by the carbon intensity, measured in grams of CO_2.

$$\text{Carbon Emissions (g } CO_2\text{)} = \text{Energy Needed} \times \text{CI}$$

By multiplying the energy needed by the carbon intensity, we effectively scale the carbon intensity value to match the amount of energy consumed. This multiplication yields the

total amount of carbon dioxide equivalent emissions associated with the energy consumption.

Despite its utility, the tool relies on average global values for power usage effectiveness (PUE) and carbon intensity, which can oversimplify the variability in energy mix and efficiency across different regions [19]. However, the methodologies and findings of Green Algorithms are fundamental to the development of our tool, serving as the base upon which we will build and make specific adjustments.

Another tool that serves a similar purpose is CloudCarbonFootprint [23]. The CloudCarbonFootprint tool focuses on cloud provider usage data, converting it into energy usage, and then applying the PUE of the data centers and the carbon intensity of the power source region [23]. This tool builds upon the "Cloud Jewels" methodology by Etsy, which estimates the energy impact of cloud computing based on virtual CPU usage, memory requests, data storage, and network traffic [24]. Due to its reliance on cloud-specific data and its emphasis on cloud environments, this tool falls outside of the scope of our research and therefore will not be used for our development.

F. Predictive Modeling and Tools for Carbon Emission Estimation

Turning now to the tools that enable us to provide estimates of future carbon emissions, we delve into approaches for forecasting carbon intensity data across different regions. We choose to forecast carbon intensity rather than carbon emissions directly, as we are estimating the emissions ourselves. Therefore, we do not wish to make our estimations the historical values which we will be feeding the model with.

Recent research employs deep learning for predictive modeling in environmental contexts. Gated Recurrent Unit (GRU) networks, suitable for handling time series data effectively, have shown promise in forecasting carbon emissions by learning from historical patterns [25]. This suggests the potential applicability of such networks to carbon intensity forecasting. This is complemented by the exploration of Fuzzy ARTMAP neural networks in energy time-series data. The use of such networks for forecasting has been deemed to be successful, benefiting from both unsupervised and supervised learning mechanisms to adapt to new patterns without discarding old information [26].

Yet, one tool that can achieve our desired goal, is a pre-trained model known as CarbonCast which offers an innovative solution by providing 96-hour (four days) forecasts. In the CarbonCast framework, the forecasting process is structured into two distinct tiers, each leveraging different architectures to predict grid carbon intensity accurately over an extended period. The first tier of CarbonCast employs Artificial Neural Networks (ANNs), one for each electricity-generating source within a region, to forecast the hourly electricity production for the next 96 hours. These ANNs use historical electricity production data to capture trends and patterns specific to each energy source, such as solar, wind, or fossil fuels. Additionally, this tier incorporates 96-hour weather forecasts, which are crucial for predicting fluctuations in renewable energy production. For instance, the expected solar or wind output can vary significantly based on weather conditions, and integrating these forecasts helps in adjusting the predictions to reflect these variabilities more accurately [12].

The second tier of CarbonCast utilizes a CNN-LSTM architecture to now generate carbon intensity forecasts. This tier takes the individual source production forecasts generated by the first tier and combines them with both the 96-hour weather forecasts and historical carbon intensity data. The CNN component of the architecture helps in extracting high-level features from the time-series data, capturing short-term temporal dynamics effectively. Following this, the LSTM layers process these features to understand and predict the long-term temporal patterns in the data. This combined approach enables the model to produce a 96-hour forecast of carbon intensity for the electricity grid, taking into account both the predicted electricity production and historical trends [12].

By leveraging these two tiers, CarbonCast can provide accurate forecasts of carbon intensity, aiding us in making informed decisions about energy use, scheduling, and carbon footprint management across various sectors. When compared with the other state-of-the-art approaches in their paper, CarbonCast was able to achieve an average decrease of 14.38% in the Mean Absolute Percentage Error (MAPE), with an average MAPE of 9.96% across all of their test regions [12]. Regardless of its performance, it is important to note that integrating the model into our analysis may require significant efforts in terms of retraining and adapting the model to accommodate specific data inputs and operational contexts unique to our study.

As we aim to inform individuals and corporations about the most carbon-efficient times and locations for running computational tasks, the methodologies and models discussed in this section, especially Green Algorithms and CarbonCast, provide us with the means for developing our solution. We will utilize the methodologies of these tools to start developing our own by adapting and integrating their findings into our proposed solution.

III. Methodology

Building on the theories and empirical grounds discussed earlier, our methodology focuses on the implementation of the tool which leverages both real-time and forecasted data to mitigate the carbon footprint of computing tasks. As we delve into the methodology, we aim to answer several questions that link directly to the environmental impacts of computing. These questions guide our developmental strategy and will help us in structuring our approach to reducing carbon emissions:

- What is the effect of Carbon Intensity on the Carbon Footprint of Computing and how can we visualize it?
- According to what indicators should we schedule our computational tasks to align with low carbon emission periods?
- How can we perform such analysis for future timeframes?

To address these, we have developed an approach that encompasses the following strategies:

Firstly, we have developed a Carbon Emission Estimation Dashboard tool that integrates real-time energy mix data, server hardware specifications, and real-time and forecasted carbon intensity data. This integration allows for a best-effort estimation of the carbon footprint for various computing tasks based on picked configurations and settings. Additionally, the dashboard includes visualizations to enhance data accessibility and comprehension. These visualizations will display the carbon footprint in relation to the time of day, hardware configurations, and geographic locations, helping users to make decisions based on clear insights. Lastly, we have utilized the CarbonCast forecasting model within the tool. This model generated forecasted carbon intensity data, allowing users to view and schedule their computational tasks during periods of expected low carbon intensity. The following sections will explore the technical architecture of the tool in detail, the data processing and cleaning performed, as well the specific models and algorithms used.

A. *Carbon Footprint Estimator Tool Design*

The Carbon Footprint Estimator Tool developed as a web application, has been designed to inform users about the environmental impact of computing tasks. This section explains the architecture and workflow of the application.

1) Overview: In the developed Carbon Footprint Estimator tool, users are greeted with a dashboard designed to simulate the environmental impact of computing operations. The interface allows users to tailor hardware configurations, adjusting parameters such as the processor model and the load level. While these parameters are different than the ones used by Green Algorithms, [19] we have modified the way we calculate carbon emissions as we will see later in this section so as to target only server CPUs. Adding on, users can set the time and duration for which they intend their computation to run for.

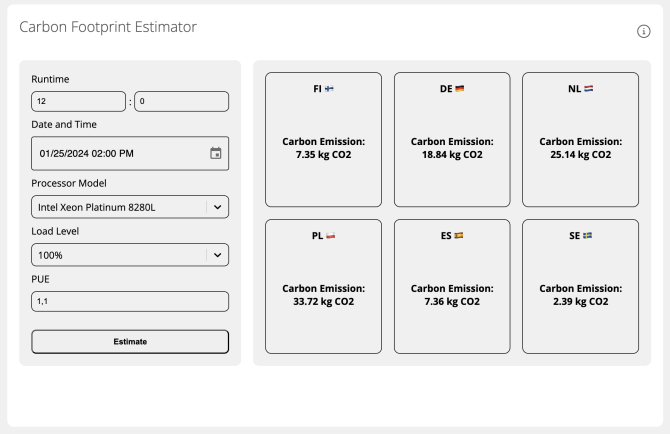


Fig. 1: User interface of the Carbon Footprint Estimator showing real-time calculation of carbon emissions for different processors and locations.

Upon configuring the desired system specifications and runtime, the tool generates the estimated carbon emissions of a task running with the selected parameters for our six different regions (Netherlands, Germany, Finland, Sweden, Spain, Poland). Alongside the estimations, hourly carbon emission data of the specification are generated (Figure 2). The graph provides a granular view of the carbon emissions generated by the computational task of Figure 1 for each hour of operation. This real-time emission tracking is essential for understanding the environmental impact of computational activities at different times of the day. The hourly data aggregation reveals patterns that may correspond with the varying carbon intensity of the power grid throughout the day. For instance, emissions may spike during peak hours when fossil fuel-based power plants are often utilized to meet the increased energy demand, while they may reduce during off-peak hours when the demand is lower, and renewable energy sources can suffice the energy needs.

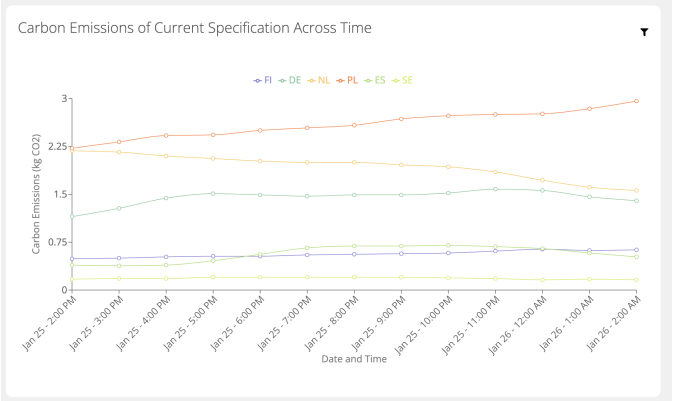


Fig. 2: Hourly Emissions Produced by Computational Task.

The other set of graphs provided by the tool extrapolate these insights further by projecting what the carbon emissions would have been had the computational task been run at any given hour in the past 24 hours, or on the same hour across the previous 30 days, or during the previous year (Figure 3). This historical analysis allows users to distinguish temporal patterns and identify the most carbon-efficient times for running their computational tasks.

As mentioned in the previous sections, the tool additionally offers forecasting capabilities which will be showcased separately in this section later on. We now elaborate on the architecture and design of the system that has led us to this developed tool.

2) System Architecture & Design: The presented architecture and design of the web application represent a developed system of components; not a pipeline. This means that data need to be manually added to the database rather than being fetched and cleaned dynamically every day.

At a high level, the system operates through a simplified flow of user interactions, data processing, and result dissemination, as depicted in Figure 4. Users interact with a frontend interface, where they can input or modify parameters. This input is then processed through a series of layers including the

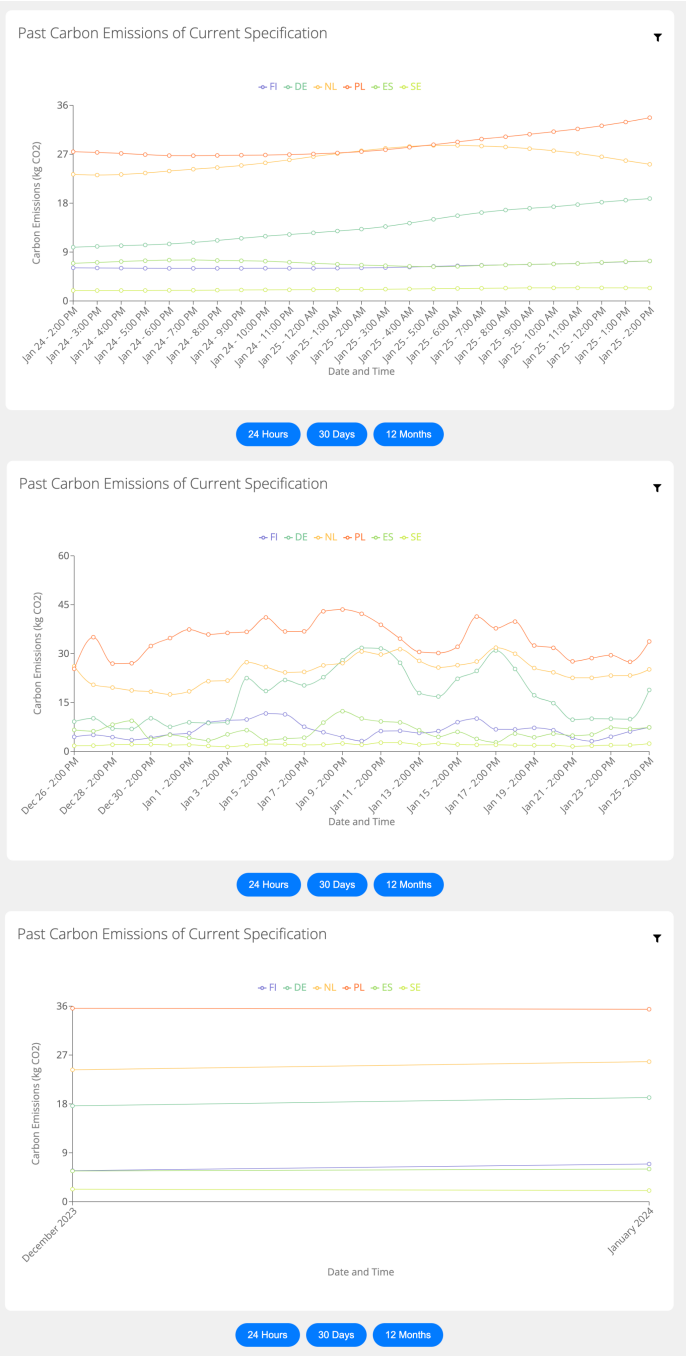


Fig. 3: Carbon Emissions of the Hardware Specification across different Time Ranges.

Business Logic Layer, and the Data Access Layer, ultimately interacting with the database for retrieval or storage of data.

Digging deeper into the architecture of the system, the web application has been designed using a three-layered architecture. In doing so, we create a modular and maintainable system as we separate the user interface, application logic, and database management into distinct layers [27]. The application consists of three layers: a Presentation Layer, a Business Logic Layer comprising an API and the Backend, and a Data

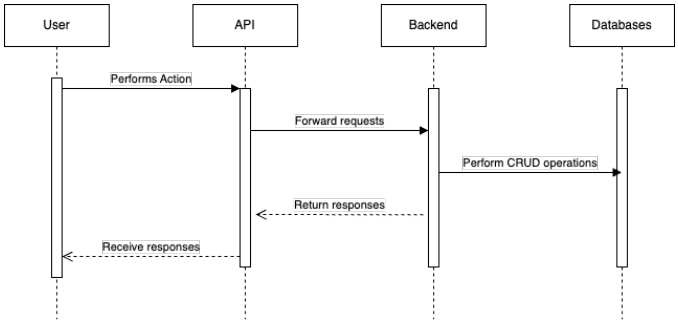


Fig. 4: Data Flow Diagram of Carbon Footprint Estimator.

Access Layer. The detailed system architecture, utilizing the three-layered structure, is illustrated in Figure 5. This diagram provides a visual breakdown of the different layers and their interconnections within the system.

The first layer, known as the Presentation Layer, encompasses the user interface of the system. The system's frontend is constructed with the Next.js framework, which offers powerful tools for building efficient and interactive web user interfaces. Next.js, built on top of React.js, gave us the ability to create dynamic and responsive web interfaces, ensuring an enhanced user experience.

Next.js capitalizes heavily on server-side rendering (SSR) to increase the websites' performance. Leveraging SSR ensures that the initial page load is efficient, providing a swift user experience as the data is already fetched before the page has been rendered. As a result, the user can begin using the page's functionality instantly, as there is no waiting time. Other than being responsible for generating a dynamic UI for the user, in this tier, user inputs are sent through API calls to the business logic.

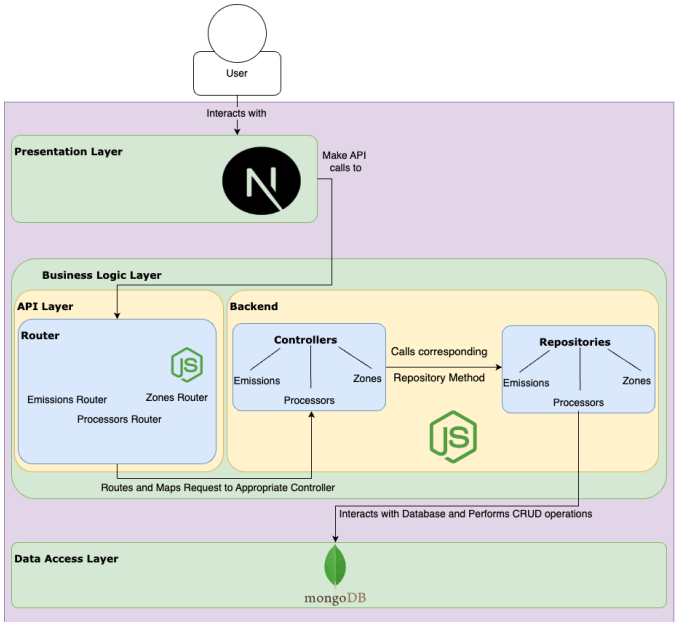


Fig. 5: High-Level Architecture of Carbon Footprint Estimator.

Here, the Business Logic Layer combines the API and the Backend. It forms the core of the system's interaction handling and business logic processing. The API, crafted using Node.js, manages the routing of requests to the appropriate controllers within the Backend. The Backend itself contains the application's business rules and logic, managing calls to the appropriate repositories that perform direct interactions with the data layer.

The third layer of our system is the Data Access Layer, which uses MongoDB. MongoDB is a flexible NoSQL database, used to manage data storage and retrieval. This layer handles all database interactions, performing CRUD operations as required by the application logic. The separation of the database management into its own layer allows for better data integrity and scalability.

This three-layered architecture not only supports the current functionalities of the carbon footprint estimator but also lays a solid foundation for future enhancements. Through this architectural approach, the tool efficiently processes user inputs, computes carbon footprints, and delivers insightful visualizations and forecasts, ensuring high performance and user responsiveness. With the architecture of the web application outlined, the next step is to examine the data used and the cleaning processes applied to ensure the tool functions effectively.

B. Data Cleaning

The data we are working with consists of hourly carbon intensity data of the following regions: Netherlands, Germany, Finland, Sweden, Spain, and Poland. However, as carbon intensity data are not easily accessible directly, we will be calculating them using the theory that was outlined in Section II-C. For this reason, we sought to collect the amount of electricity generated from every source our regions use, and after that utilize their corresponding emission factors to calculate the carbon intensity of a region at a particular time.

The data cleaning process involved collecting and processing granular electricity generation data from the European Network of Transmission System Operators for Electricity (ENTSO-E) Transparency Platform [28], which serves as a source of power generation data across Europe. ENTSO-E provides access to real-time and historical data concerning electricity production and consumption across member countries.

The raw data from ENTSO-E includes yearly electricity generation data categorized by production types, such as biomass, various forms of fossil fuels, nuclear, and renewable sources such as solar and wind. Now, depending on the regional reporting standards, data are recorded at 15-minute, 30-minute, or even hourly intervals, indicating how much energy has been generated within this interval. During our data cleaning, we began by importing multiple datasets for the selected regions, covering different years. Once imported, each file was read into a DataFrame, where we stripped any unnecessary quotes from column headers and data. In this way, we ensured that there was consistency across all data points. Then for each region, we addressed all the entries marked as 'n/a' (not available) or 'n/e' (not expected), which are common placeholders in the data. 'n/a' values are converted to NaN to indicate missing data, while 'n/e' are treated differently, by being set to zero, reflecting the expected absence of generation from those sources at certain times.

After the initial cleaning, we standardized the data by merging similar energy sources under some predefined categories. More specifically, ENTSO-E collects generation data for each region for the following sources of energy:

- Biomass
- Fossil Brown coal/Lignite
- Fossil Coal-derived gas
- Fossil Gas
- Fossil Hard coal
- Fossil Oil
- Fossil Oil shale
- Fossil Peat
- Geothermal
- Hydro Pumped Storage
- Hydro Run-of-river and poundage
- Hydro Water Reservoir
- Marine
- Nuclear
- Other
- Other renewable
- Solar
- Waste
- Wind Offshore
- Wind Onshore

We now aggregated the data into broader categories using a predefined mapping as shown below:

- **Biomass** encompasses direct entries from Biomass.
- **Coal** is made up of the aggregation of the Brown coal/Lignite, Hard coal, Peat, and Coal-derived gas columns.
- **Natural Gas** is captured under a single category, simplifying all entries from Fossil Gas.
- **Geothermal** gets mapped to itself.
- **Hydro** is made up of the energy source data of Pumped Storage, Run-of-river and poundage, and Water Reservoir.
- **Nuclear** energy, is tracked as it is.
- **Oil** includes both standard Oil and Oil shale.
- **Solar** energy, is distinctly categorized.
- **Wind** encompasses both Offshore and Onshore.
- **Unknown** or unspecified energy sources include Marine, Waste, and Other non-renewable sources that do not fit into the other categories.

This standardization is crucial as it directly ties into how we apply carbon emission factors later in our analysis. This categorization simplifies the application of specific carbon emission factors, which vary across different energy types.

After forming a new DataFrame with just these new categories, we proceeded with resampling the data to an hourly format, regardless of the original granularity. During this

resampling, we aggregated the data using a summing method for each hour. It is important to note that our aggregation method specifically excludes NaN values from the calculations. If an hour's data consists entirely of NaN values, indicating a complete lack of reported data for that period, the aggregated output for that hour is set to zero, reflecting an absence of energy generation. The final step of our cleaning involved calculating the hourly carbon intensity value. Each category like 'biomass', 'coal', and others, is associated with the emission factors of Table I that reflect the typical emissions produced per unit of electricity generated by that source. By consolidating the data in this manner, we were able to use Formula 1 and as a result, calculate the data we desired per hour and location.

C. Estimation of Carbon Emissions

As discussed in Section II-E, by having the carbon intensity of a region at a particular time, we can estimate the carbon emissions of a computational task, given we know how much energy that task requires to run. To do that, we used the foundations established by Green Algorithm's methodology, however, tailoring them to our needs, as we are solely dealing with server operations. In order to figure out how much energy a computational task ran on server components generated, rather than using variables such as the power draw of memory and processor, core utilization factor, and the number of cores, we transitioned into using the metrics and valuations provided by the SPEC Power benchmark.

1) SPEC Power Benchmark: The SPEC Power Benchmark, developed by the Standard Performance Evaluation Corporation (SPEC), is a benchmark which evaluates the power and performance of servers. It provides a measure of energy efficiency and assesses how much work a server can perform in relation to its power consumption. More specifically, the SPECpower_ssj2008 benchmark, which is what we will be using for this project, simulates typical server-side operations using different workloads. By doing so, it measures the power consumption at various load levels which can range from idle to 100%. In this study, we utilize the results from SPECpower_ssj2008 to determine the energy consumption of our computational tasks based on the energy required to run these tasks on server components [29].

In our dashboard, users can select from several server CPUs and specify the desired load level to estimate the energy consumption. For instance, let's consider the SPECpower_ssj2008 results for the ASUSTeK Computer Inc. RS720-E9-RS8, which has an Intel Xeon Platinum 8280L CPU. When put under testing, the CPU demonstrated different power consumption values at various load levels, as illustrated in the provided benchmark results of Figure 6. Users can select the load level, such as 50%, which corresponds to an average active power of 187 watts. This value represents the energy needed by the system at the specified load level. By using this information, the dashboard retrieves the total energy consumption and subsequently estimates the carbon emissions based on regional carbon intensity data.

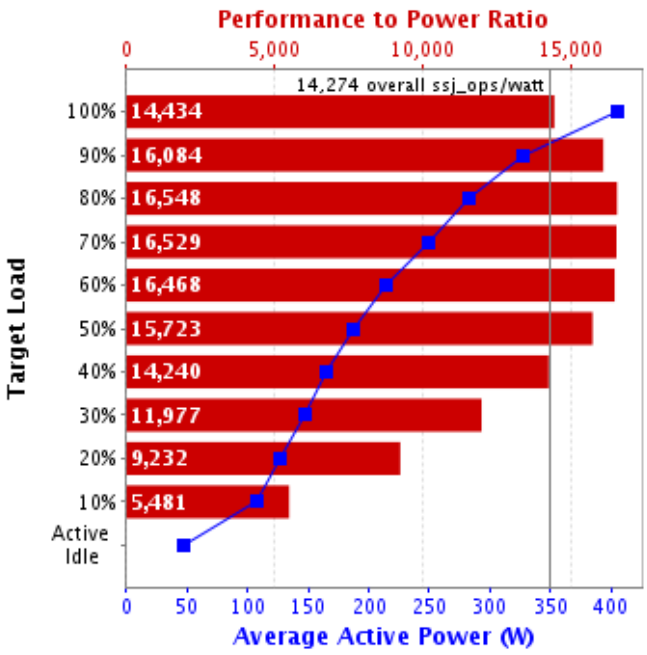


Fig. 6: SPECpower_ssj2008 Benchmark Results Summary for ASUSTeK Computer Inc. RS720-E9-RS8.

2) Calculating Server Emissions: Having said all this and building on the work of Green Algorithms, we estimate the carbon emission of a computational task run on a server for a single instance as follows:

$$\text{Total Power} = \text{Power at Load} \times \text{PUE} \times 0.001 \quad (2)$$

Here, *Power at Load* is measured in watts (W), representing the power usage at a given load level. This variable is multiplied then by the PUE, to account for the additional power needed to operate the data center infrastructure as explained in Section II. Lastly, the multiplication by 0.001 converts the power from watts to kilowatts (kW), aligning the units for the next step in the calculation. Thus, now that we multiply the *Total Power* with the *Carbon Intensity* which is expressed in grams per kilowatt-hour (g/kWh) the resulting emissions are calculated in grams of CO_2:

$$\text{Carbon Emissions} = \text{Total Power} \times \text{Carbon Intensity} \quad (3)$$

After calculating the energy needed and its associated carbon emissions for a single instance of computational activity, we estimate this process for the entire duration of the task's runtime. For instance, if a computational task is scheduled to run for 12 hours, we repeat the multiplication of energy consumption with the carbon intensity for each hour within that duration. The user selects the start date and time for executing the algorithm, and from that point onward, the algorithm continues for the specified runtime. For example, if the algorithm starts execution in Finland on a specific date at 10:00 AM and runs for 12 hours, we obtain the carbon

Algorithm 1 Calculate Total Carbon Emissions for Computational Task

Require: processorName ▷ Name of the processor used for the task
Require: loadLevel ▷ Operational load level of the processor
Require: PUE ▷ Power Usage Effectiveness of the data center
Require: time ▷ Start time of the computational task
Require: runtime ▷ Duration of the task in hours
processorDetails ← getProcessorDetails(processorName) ▷ Retrieve details of the specified processor
powerAtLoad ← getPowerUsageForLoad(processorDetails.specpower_ssj2008.load_power, loadLevel) ▷ Determine power usage at given load
totalCorePower ← PUE × powerAtLoad ▷ Calculate total power accounting for data center inefficiency
energyNeeded ← totalCorePower × runtime / 1000 ▷ Convert power to energy consumed over runtime, from watts to kilowatts
carbonIntensityData ← findCarbonIntensityData(time, time + runtime) ▷ Retrieve carbon intensity data for the duration of the task
emissionsList ← [] ▷ Initialize list to store emissions per hour
for each ci in carbonIntensityData **do**
hourlyEmissions ← energyNeeded × ci.carbonIntensity / 1000 ▷ Calculate emissions for each hour based on carbon intensity
Append hourlyEmissions to emissionsList
end for
totalEmissions ← sum(emissionsList) ▷ Sum all hourly emissions
return totalEmissions ▷ Return the total emissions for the computational task

intensity values for each hour in Finland during that period. We then multiply the energy consumption for each hour by the respective carbon intensity value and add everything together, giving us the total emissions across all hours. Something to note here is that *Total Power* is assumed constant throughout the duration the algorithm runs. This assumption was necessary due to limitations in data availability and the difficulties that arose with capturing dynamic changes in power supplied to server CPUs at an hourly granularity. We can now repeat the same process for multiple countries, by utilizing that country's carbon intensity data which are different at different points of the day. This iterative process, further described in Algorithm 1, enables us to accurately capture the temporal and spatial variations in carbon emissions and provides a comprehensive assessment of the environmental impact of the computational task.

This calculation enables us to quantify the environmental impact of computational tasks and contributes to efforts to mitigate climate change. By gaining insights into the carbon footprint of computing activities, individuals and organizations can make informed decisions to reduce their environmental impact and shift their behaviors by scheduling their tasks during non-intensive intervals.

D. Running and Validating the CarbonCast Forecasting Model

So far we have discussed about estimating carbon emissions for past time frames as our data do not span future times. However, it is important to be able to generate carbon emission estimates also in the future, to further give users the ability to strategically decide in what region and time they wish to deploy their tasks on the server.

In our project, the implementation of the CarbonCast forecasting model was central to predicting carbon intensity. The process was divided into three distinct phases: obtaining weather data, processing this data, and running the forecasting model itself.

Before delving into the steps we used for integrating the model, it is important to mention that our forecasts are limited only to Germany. This stems from the need to fetch new weather data. The original model was developed and trained with weather data from 2019 to 2021. Thus as we wished to generate forecasts for 2024, we had to acquire more recent data to retrain the model and achieve more accurate results. Retrieving data of such high volume (hourly data from 2020 to 2023) proved more challenging and time-consuming than anticipated. Consequently, dealing with this issue for all of our regions would have been impractical within our project's timeframe. Therefore, we focused exclusively on Germany, where we could manage the data acquisition and processing within the available period. The same approach though followed can be applied to any country as long as the required data have been acquired.

1) Weather Data Acquisition and Processing: The first step in generating the 96-hour carbon intensity forecasts revolved around retrieving the weather forecasting data. The weather data that we sought are 96-hour forecasts of variables such as wind speed, temperature, dewpoint, downward short-wave radiation flux, and precipitation. We collected the data via the GFS weather forecast archive [30]. After registration and configuring the archives' configuration files, we were able to fetch data tailored to our regional and temporal specifications in grib2 format. Due to this format, we faced significant

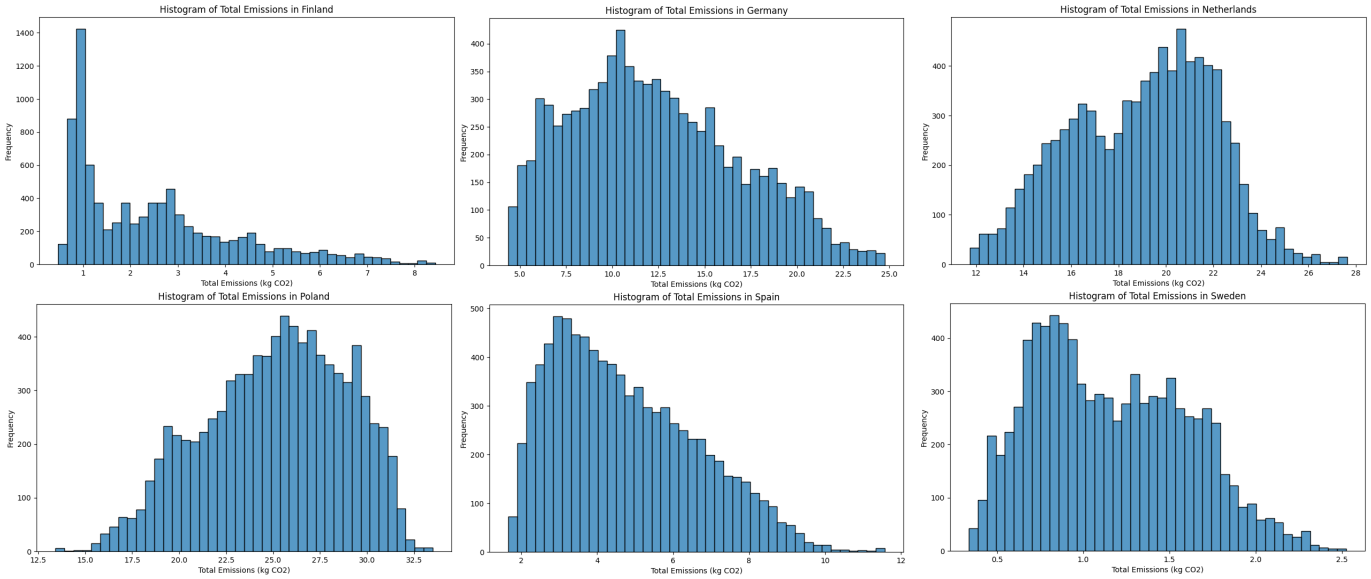


Fig. 7: Distribution Plots of Total Carbon Emissions of Computational Task across Regions.

difficulties parsing the data. In order to parse grib2 files, the wgrib2 library was required. Downloading the wgrib2 library proved to be a cumbersome task. As we were developing our system using the latest MacBook model that runs on the M1 processor, it was evident that the library was not yet configured to be used in such developing environments. Key challenges included managing dependencies via Homebrew. This involved locating the correct directory paths for Homebrew installations on Apple Silicon, which differ from the paths used on Intel Macs. Furthermore, having the correct version of GCC proved to be crucial, as the default GCC compiler version installed by Homebrew was not compatible with wgrib2. This necessitated the manual installation of a specific, compatible version of the GCC compiler to avoid compilation errors. Additionally, the cmake configuration for wgrib2 was modified to download third-party libraries necessary for its compilation, rather than merely attempting to import them. This step was crucial because direct imports were failing due to compatibility issues with the libraries on the new Silicon architecture.

2) Configuration and Model Execution: Once the necessary data was integrated with our existing datasets, changes were made to the configuration settings of the model to align the data correctly for reading. These changes ensured that the data inputs were properly accessed by the different tiers and in the format expected. After that, we ran the model's first tier to predict source production. Following the first tier's successful completion, we went on to the second tier, which made use of the first tier's data to forecast carbon intensity over 96 hours. At this point it was decided to retrain the model using our latest data. This ensured to us that more accurate data were generated as we will also see in Section IV.

A note to be made is that the data that the model forecasts are test data. This means that it generates data up until the timeframes of the inputted data so they can be then compared against the actual values. In its current iteration, the model generates such data to replicate the results of the researcher's paper. For that reason, we created an additional script, which by utilizing the architecture and the model of CarbonCast, generated 96-hour carbon intensities without a dependency on the inputted timeframes. In this way, we were able to acquire forecasts that exceeded the periods outlined in our datasets.

Throughout this process, we worked closely with the model's developers to make sure that any differences in the predictions were caused by the model's implementation rather than mistakes in our application. This was crucial to verifying the changes we made in the configuration files and guaranteeing that the predictions were as accurate as the model could produce.

IV. Analysis

Now that we have introduced the tool and presented its functionality, we will analyze the results that the user of the dashboard can record. The randomly selected computational task that we will explore will run for 12 hours, with a configuration that utilizes the server CPU, Intel Xeon Platinum 8280L, a PUE of 1.1, and a load of 80% being supplied to it.

A. Influence of Energy Mix and Time on Carbon Emissions

The graphs above (Figure 7) have been generated by modeling the use of the described computational task that begins execution at each hour of the day for 12 hours. The data span from April 2023 to April 2024, where the same configuration was used for all six countries in question. Through these carbon emissions histograms, we will examine the geographical disparities in carbon intensity across these locations and discuss the energy mixes and operational efficiencies in each region.

Country	Date	Footprint at 14:00 PM	Carbon Intensity at 14:00 PM	Footprint at 02:00 AM	Carbon Intensity at 02:00 AM
Finland	5-6th, Mar	0.43	115.29	0.45	120.07
Germany	5-6th, Mar	1.30	347.58	1.62	432.52
Netherlands	5-6th, Mar	1.65	440.55	1.60	428.67
Poland	5-6th, Mar	2.15	574.30	2.37	633.11
Spain	5-6th, Mar	0.14	37.34	0.32	85.22
Sweden	5-6th, Mar	0.11	30.73	0.11	30.07
Finland	7-8th, Mar	0.48	129.51	0.50	132.98
Germany	7-8th, Mar	0.92	246.53	0.98	261.33
Netherlands	7-8th, Mar	1.69	453.14	1.18	315.41
Poland	7-8th, Mar	1.60	428.09	2.41	644.06
Spain	7-8th, Mar	0.11	30.65	0.17	45.07
Sweden	7-8th, Mar	0.14	36.78	0.11	29.94

TABLE II: Carbon Emissions and Intensity Across Different Times, Days and Locations. The emissions are given in kg CO_2 and the carbon intensity in gCO_2/kWh.

Beginning our analysis, we can immediately observe a pattern that ties the type of energy sources used to the environmental impact of running computational tasks on servers. For instance, Sweden and Finland demonstrate lower carbon emissions for the whole task, with more of them being clustered below 1.5 kg CO_2 and 2.5 kg CO_2 per computational task respectively. Specifically, the 75th percentile for Sweden is 1.51 kg CO_2, indicating that 75% of the emission values are below this level. Similarly, for Finland, the 75th percentile is 3.39 kg CO_2, suggesting a slightly higher emission range but still demonstrating that the majority of emissions are lower, with a significant number falling below 2.5 kg CO_2. This indicates a predominantly clean energy mix, reflecting a high utilization of renewable energy sources. This can be further verified by observing each country's percentage use of renewable and non-renewable sources (refer to Appendix Figure 10). Finland's significant use of nuclear power (45.3%) along with its large use of renewable sources like wind (19.6%) and hydro (19.1%) results in a cleaner energy mix. Likewise, Sweden's mix, with a large emphasis on hydro (42.4%), nuclear (30.8%), and supplemented by wind (21.8%), showcases a similar trend of minimal carbon footprints. The direct relationships which exist between energy production to carbon intensity and as a result carbon emissions, portray how the clean energy mix employed by these countries significantly lowers the carbon intensity of computational tasks executed within their borders and therefore the associated emissions.

In contrast, Germany's mixed energy portfolio featuring significant amounts of coal (24.3%), natural gas (12%), and renewable sources such as wind (33.2%) and solar (12.7%), experiences moderate emission levels for the computational task. Germany's histogram (Figure 7(b)) shows a wider distribution of CO_2 emissions, with peaks between 12.5 to 17.5 kg CO_2 per task. The relatively high utilization of natural gas and renewables helps mitigate some of the emissions mostly associated with coal energy production. Similar to Germany, is the Netherlands. The Netherlands utilizes natural gas (28.5%) and generates most of its electricity from sources outside the ones defined in this paper. Those attribute to 37.8% of the country's energy generation, causing a skewed distribution. In the Netherlands, emission levels for computational tasks predominantly range between the 25th percentile of 16.58 kg CO_2 and the 75th percentile of 21.31 kg CO_2 with a median of 19.34 kg CO_2. This indicates a narrower, yet higher range of emissions compared to Germany. Such an argument can not be made for Poland though. Poland's energy profile is heavily dominated by coal, accounting for 62.3% of its energy production. This dependence on coal translates to higher emission levels, with computational tasks resulting in more substantial carbon footprints. The histogram (Figure 7(d)) depicts the highest emission range out of all the other countries, with computational tasks often causing emissions in the range of 20 to 30 kg CO_2, as shown by the 75th percentile reaching up to 28.08 kg CO_2.

Lastly, Spain employs a diverse mix of renewable and non-renewable energy sources, leading to moderate emission levels. In Spain, the fluctuations in solar and wind energy production suggest further stabilization in the energy grid to consistently reduce emissions. These differences in energy mixes are what cause carbon intensity to differ so much per region (refer to Appendix Figure 12).

Energy mixes tend to not change too much on a year-to-year basis unless significant environmental events or changes in energy policy occur. Therefore, the observed trends between April 2023 and April 2024 likely reflect the current trend and differences between these regions. However, observations are not static and can be influenced by changes in energy policies, technological advancements in renewable energy, or fluctuations in weather conditions that affect renewable energy production. For example, an unusually sunny or windy year could temporarily increase the output of solar and wind installations, leading to lower carbon emissions than what is typically expected. On the other hand, a particularly calm or cloudy season could increase the dependency of nations on fossil fuels, thus increasing the carbon intensity.

This can be further analyzed by looking at the values of Table II. From the values, we can observe the direct relationship between carbon intensity and carbon emissions. This relationship is evident in the changes observed in CI values from midday to early morning between the six countries. For example, in Poland, the carbon emissions at 14:00 PM on March 7th were 1.60 kg CO_2, which increased to 2.41 kg

CO_2 by 02:00 AM the following morning. This increase aligns with a higher CI during the nighttime affected by the changed energy mix of the country at that time. Conversely, when there is a drop in CI values, such as in the case of the Netherlands from 453.14 gCO_2/kWh to 315.41 gCO_2/kWh between March 7th-8th 2024, carbon emissions also experience a 30% decrease; as much as CI decreased.

Similarly, the variations between different days also reveal how time affects carbon emissions. In Spain, the CI at 14:00 PM on March 5th was 37.34 gCO_2/kWh, contributing to emissions of 0.14 kg CO_2, whereas on March 7th at the same time, the CI decreased to 30.65 gCO_2/kWh, and the emissions dropped slightly to 0.11 kg CO_2. These fluctuations highlight how dynamic changes in the energy mix, influenced by factors such as weather or operational adjustments in the power grid, directly affect the CI and subsequently the carbon emissions. The data in Table II and Table III, showing the minimum and maximum emissions of the whole computational task, not only reinforces the understanding of how location influences carbon emissions through different energy mixes but also shows the critical impact of temporal factors. Observing CI and emissions at different times of the day provides clear evidence that timing computational tasks to coincide with periods of lower CI can effectively reduce the carbon footprint of these operations (Figure 8b).

Country	Mean	Median	Q1	Q3	Min	Max
Germany	12.31	11.79	8.80	15.38	4.37	24.75
Netherlands	18.99	19.34	16.58	21.31	11.74	27.63
Poland	25.14	25.45	22.49	28.08	13.42	33.50
Spain	4.75	4.43	3.21	6.07	1.67	11.57
Sweden	1.16	1.11	0.80	1.51	0.34	2.53
Finland	2.50	2.11	1.00	3.39	0.47	8.43

TABLE III: Statistical Summary of Carbon Emissions for Computational Tasks across Various Countries (April 2023 - April 2024). The emissions are given in kg CO_2.

1) Statistical Analysis of Regional Carbon Intensity Differences: In order to quantify these differences, we seek to explore if there are significant differences between the carbon intensities recorded in each region. To do that, we wish to see if there exists a statistical significance between the carbon intensities recorded in each of the six countries. The simplest way to do such a test is via an ANOVA (Analysis of Variance) test, a statistical test used to analyze the difference between the means of more than two groups. To continue with this though, we must ensure our data meet the assumptions laid out for performing an analysis of variance.

For us to proceed with this test, the carbon intensity of each country must be normally distributed, and there must be homogeneity of variance. The assumption regarding the independence of groups is already satisfied, as we are comparing values from different countries. Therefore, we must ensure the other two assumptions are fulfilled before continuing. Starting with the homogeneity of variance, we see immediately how the assumption is violated in Table IV. None of the variances between the six countries are equal or close to each other.

Country	Mean	Median	Variance	Skewness
Germany	250.63	240.45	11376.23	0.36
Netherlands	389.21	394.08	5014.08	-0.16
Poland	515.43	530.20	10038.74	-0.45
Spain	96.73	83.52	2121.85	0.72
Sweden	23.81	22.94	88.78	0.31
Finland	51.72	41.60	1339.43	1.11

TABLE IV: Statistical Summary of Carbon Intensity Across Different Countries from April 2023 to April 2024.

When it comes now to the normality of the data, the mean, the median, and the skewness can all tell us something regarding the distribution of our values. When the data is perfectly normal, the mean and the median are identical. In our case, the values deviate from one another by at most 15 gCO_2/kWh. This is further depicted by the skewness of our data, where none of the values are close to 0. Due to this, there is strong evidence that the data are skewed and thus non-normal. To verify this before continuing with a non-parametric test, we have graphed Q-Q plots for all of the countries (refer to Appendix Figure 11) and have conducted a Shapiro-Wilk test. In all the Q-Q plots, the data points significantly deviate from the red line (which represents a perfect normal distribution), especially in the tails. This suggests that the carbon intensity data for these countries do not follow a normal distribution. Moreover, all p-values of the Shapiro-Wilk test are smaller than 0.05.

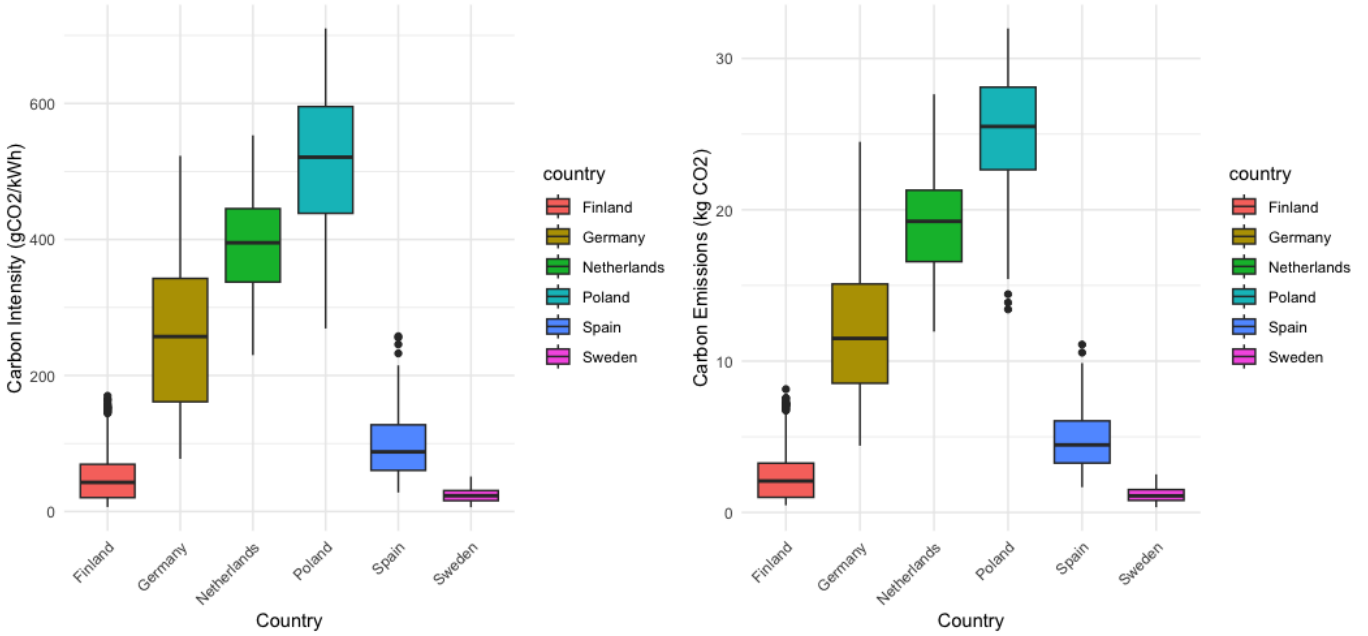
Country	W statistic	p-value
Sweden	0.96782	4.49×10^{-14}
Netherlands	0.98794	2.53×10^{-7}
Finland	0.88775	2.57×10^{-26}
Spain	0.93609	2.74×10^{-20}
Germany	0.96336	3.73×10^{-15}
Poland	0.97276	9.47×10^{-13}

TABLE V: Results of the Shapiro-Wilk tests for normality of carbon intensity data.

Thus, the null hypothesis (H_0) of the test, stating that the data come from a normal distribution is rejected. Having reached this conclusion, we look at non-parametric tests to perform our statistical significance analysis. A test for which our data meets the assumptions placed is the Kruskal–Wallis test. Kruskal–Wallis is used to compare two or more groups, but now we compare the medians rather than the means. In other words, it tests whether the mean ranks are the same across the countries. Based on this, our hypothesis is as follows:

- H_0: The carbon intensity medians are equal across all countries.
- H_1: The carbon intensity medians are not equal across all countries.

After performing the test, we find a p-value of 2.2×10^{-16}, which is smaller than the theoretical accepted value of 0.05. As a result, we reject our null hypothesis and thus the carbon intensity medians were shown to not be equal. The below box plots further showcase this point (Figure 8a).



(a) Boxplot of Carbon Intensity by Country. This plot shows the distribution of carbon intensity (gCO_2/kWh) for different countries from April 2023 to April 2024.

(b) Boxplot of Carbon Emissions of the Complete Task by Country. This plot shows the distribution of carbon emissions (kg CO_2) for different countries from April 2023 to April 2024.

Fig. 8: Comparison of Carbon Intensity and Footprint by Country.

We observe that the box plots for Germany, the Netherlands, and Poland do not overlap, indicating a clear difference in carbon intensity between these countries. Although there is a slight overlap between the box plots of Finland and Spain, as well as between Finland and Sweden, the medians of these groups do not lie within the overlapping regions. Consequently, this still suggests that there is a significant difference in carbon intensity among these regions.

In this way, we conclude that there is a significant difference between these countries when it comes to their carbon intensity values and as a result, the energy mixes that they are utilizing. Such a result further quantifies what has been established through observation in this section, that location and time have significant effects on the estimation of carbon emissions in all sectors even in computing.

2) Relationship between Carbon Intensity & Carbon Emissions: In developing our tool, to estimate the carbon emissions we used the simple Formula 3. An assumption we had made regarding the formula was that *Total Power* remains constant throughout the duration that the algorithm runs. This simplification implies a direct linear relationship between carbon emissions and carbon intensity, as carbon intensity is the only variable affecting emissions during the computation interval. Due to this, the model itself is not accurate and does not account for real-world complexities where power usage can vary. However, as the analysis so far has shown, through the model, the influence of varying electricity sources on carbon intensity can be observed. Using our analysis but also the established literature, it can be deduced how reduced emissions in computing tasks are linked to low carbon intensity periods; meaning periods where the energy mix is cleaner. Thus, while our model simplifies the relationship, it highlights the linkage between energy sourcing, carbon intensity, and resultant emissions, showcasing the importance of energy source management in environmental impact strategies (Figure 8).

B. Analysis of Forecasting Model Performance

We now transition to the analysis related to the results retrieved using the CarbonCast forecasting model. This section evaluates the model's performance before and after training and analyzes the discrepancies between forecasted and actual carbon intensities.

To begin with, by using the model we were able to generate test forecasting data ranging from the 23rd of November, 2023 until the 28th of May, 2024. The data include for every day, 96-hour carbon intensity forecasts. This data was generated with the pre-trained model and then once again with the newly retrained model. The improvement in model performance after training is evident when comparing the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) values across four consecutive days. RMSE measures the error of a model in predicting quantitative data. It is calculated as the square root of the average of squared differences between predicted and actual observations. On the other hand, MAPE measures the accuracy of a model as a percentage. Initially, the model exhibited an overall RMSE of 118.31 and a MAPE of 56.78%, values indicating large variance and error in the predictions compared to the actual data.

Day	RMSE	MAPE
Day 1	94.43	42.92%
Day 2	120.99	58.22%
Day 3	128.04	62.53%
Day 4	129.78	63.44%
Overall	**118.31**	**56.78%**

TABLE VI: Performance of the Forecasting Model Before Training.

However, after retraining the model with the up-to-date data, these metrics improved significantly, with the overall RMSE reducing to 82.64 and the MAPE to 32.86%.

Day	RMSE	MAPE
Day 1	70.75	25.41%
Day 2	86.79	34.34%
Day 3	86.46	35.89%
Day 4	86.55	35.80%
Overall	**82.64**	**32.86%**

TABLE VII: Performance of the Forecasting Model After Training.

Thus, as we were certain that the retraining improved the predictions, we continued by utilizing the 96-hour forecasts that we generated with our script. These can be seen being plotted against the actual values of these days for Germany in Figure 9. By observing the graph, it is clear that the model does a very good job of capturing hourly and daily trends with high accuracy. Any fluctuations in the intensity values of these four days have been captured correctly, despite some overshooting or undershooting that may take place from time to time. Such small inaccuracies are expected, as it would be impossible for the model to capture effectively any unforeseen spikes in energy demand or sudden changes in renewable output. As a result, by utilizing such efficient forecasts of carbon intensity, we can perform estimations for the carbon emissions of computational tasks also in future timeframes. In this way, users can be given the ability to strategically deploy their computational tasks at locations where the emission output will be minimized.

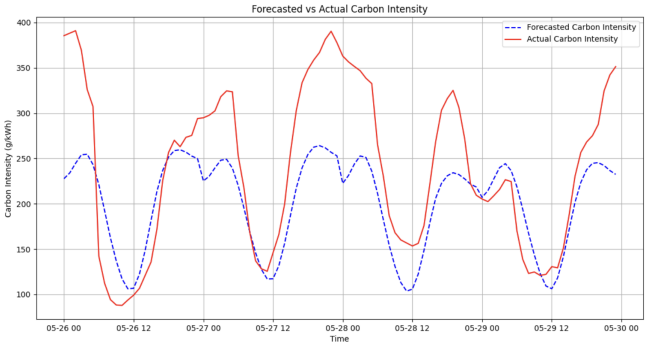


Fig. 9: Comparison of Forecasted vs. Actual Carbon Intensity.

V. Discussion and Future Work

The analysis conducted in Section IV demonstrated that carbon intensity directly affects the carbon footprint of computational tasks. More specifically, a higher carbon intensity, caused by a larger use of fossil fuels in a region's energy mix, results to the increased emissions of computational operations. Conversely, lower carbon intensity, reflective of a cleaner energy mix, reduces these emissions. By integrating real-time and forecasted carbon intensity data into our tool, we provided users with the ability to visualize such effects that their computing activities may have. The visualizations not only portray the variance in emissions due to different energy mixes but also showcase the reduction that can take place when computational tasks are scheduled during periods of lower carbon intensity.

The ability to schedule computational tasks during periods of low carbon intensity is critical for reducing carbon emissions. Our findings suggest that scheduling these tasks should occur when the energy mix is at its cleanest, meaning when carbon intensity levels are lowest. An alternative is to schedule these tasks in regions where the energy mix is predominantly composed of renewable sources (e.g. Finland and Spain).

To schedule tasks in alignment with periods of low carbon intensity, however, it is necessary to know what the intensity levels are beforehand. This necessity brings forecasting into a pivotal role. Our integration of the CarbonCast forecasting model enables the prediction of carbon intensity up to four days in advance, providing a crucial planning tool for users. By forecasting future carbon intensity, the tool allows users to be informed regarding where and when they should deploy/schedule their computational tasks maximizing their alignment with environmental and operational efficiency goals.

The development of our dashboard has provided valuable insights into carbon emissions from computational tasks across different geographic locations. However, the current architecture of our system as individual components rather than a pipeline hampers the usability of the system in real-time scenarios. Therefore, a major step that has to take place for future iterations, is the transformation of the current system into a deployable pipeline. This would involve setting up a scheduled polling mechanism that retrieves new data every 24 hours and generates updated forecasts. Implementing this pipeline would ensure that our dashboard can be used dynamically and can provide real-time insights and forecasts, allowing users to make timely and informed decisions about their computational tasks and energy usage strategies.

Now, when it comes to estimating the emissions, one of the main assumptions in our current model is the non-variability of the *Total Power* variable during computational tasks. This assumption overlooks the effects of power dynamics in computational systems. For instance, we can observe this even through the SPEC Benchmark which indicates that increasing the load on systems can lead to states of power inefficiency. Thus, considering the already known effects that exist if we change the load, it is unrealistic to assume power remains

constant across large timeframes even if it deviates by small factors. To address this, in the future, we should consider a function where *Total Power* changes across time. This would allow for a non-linear model of the form $y = a \times b$, where both a (*Total Power*) and b (*Carbon Intensity*) are continuous variables. Such an approach would enable a deeper analysis of the interdependencies between carbon intensity, computational energy use, and carbon emissions. By capturing such variability in power usage, we would be able to estimate emissions more accurately.

In conclusion, while our current system serves as a tool for understanding and managing carbon emissions, the outlined future work is vital for ensuring our solution is precise and practical. In the future, we aim to deploy and modify its architecture so it follows that of a pipeline solution. This will allow for more dynamic and responsive carbon footprint management by our users. Additionally, we plan to expand upon the current functionality, by including features that allow for a dynamic change in the task's computational load during runtime, thus increasing the tool's accuracy and applicability.

VI. Conclusion

In this thesis, we introduced a tool designed to estimate the carbon emissions of computational tasks on servers. Our Carbon Footprint Estimator integrates real-time data on energy mix and server hardware configurations along with models like CarbonCast to estimate and forecast carbon emissions. We demonstrated the ability of our tool to provide insights into the environmental impact of computational activities, highlighting the significance of timing and geographical location on carbon emissions.

Our findings reveal that scheduling computational tasks during periods of lower carbon intensity can reduce the carbon footprint of server operations. Specifically, our results indicate that by leveraging low-carbon intensity periods, the tool can help in planning and executing computing tasks more sustainably. The use of detailed real-time and forecasted carbon intensity data enables users to make informed decisions about when and where to run their computational tasks to minimize environmental impact.

Ultimately, the significance of the Carbon Footprint Estimator tool transcends its immediate functionality. It catalyzes change, by promoting awareness and driving the tech industry towards a more sustainable and environmentally-friendly future. Sustainability must be at the core of our digital evolution, ensuring that as we progress technologically, we also protect the health of our planet.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Vasilios Andrikopoulos and Prof. Brian Setz, who have supported and guided me through the completion of this thesis. Their expertise and insights have been instrumental in shaping the direction of this thesis.

Special thanks go to Glenn Schreiber from the Philadelphia Weekend Weather & Storm Forecasts blog [31]. His technical expertise and willingness to assist were invaluable, particularly his help with the installation and troubleshooting of the wgrib2 library, which was crucial for my data processing tasks.

I am also immensely grateful to Diptyaroop Maji, a Doctoral Student at the University of Massachusetts Amherst and a developer on the CarbonCast team [12]. His guidance was instrumental in ensuring that my integration of the forecasting model was seamless and error-free. Diptyaroop's readiness to answer my questions helped me deepen my understanding of the model.

References

[1] Intergovernmental Panel on Climate Change. Geneva, Switzerland: IPCC, 2023, pp. 35–115.

[2] E. de Boer, K. Ellingrud, G. Richter, and D. Swan, "What are industry 4.0, the fourth industrial revolution, and 4ir?" *McKinsey & Company*, August 2022. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir

[3] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing carbon: The elusive environmental footprint of computing," *IEEE Micro*, vol. 42, no. 4, pp. 37–47, 2022.

[4] European Commission, "Ict environmental impact (rp2024)," https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/ict-environmental-impact-rp2024, 2024, accessed: 2024-07-30.

[5] S. Chamanara, S. A. Ghaffarizadeh, and K. Madani, "The environmental footprint of bitcoin mining across the globe: Call for urgent action," *Earth's Future*, vol. 11, no. 10, 2023. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023EF003871

[6] C. Stoll, L. Klaaßen, and U. Gallersdörfer, "The carbon footprint of bitcoin," *Joule*, vol. 3, no. 7, pp. 1647–1661, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542435119302557

[7] Technology Review. (2022) We're getting a better idea of ai's true carbon footprint. Accessed: 2023-10-05. [Online]. Available: https://www.technologyreview.com/2022/11/14/1063192/were-getting-a-better-idea-of-ais-true-carbon-footprint/

[8] N. Mads, "Default emission factors," https://github.com/electricitymaps/electricitymaps-contrib/wiki/Default-emission-factors, 2024, accessed: July 9, 2024.

[9] M. Brain and D. Roos, "How power grids work," HowStuffWorks, 1970, accessed: 2024-07-09. [Online]. Available: https://science.howstuffworks.com/environmental/energy/power.htm

[10] Just Energy, "Power grid: What is it and how does it work?" https://justenergy.com/blog/power-grid-what-is-it-and-how-does-it-work/, December 2021, last updated on April 8, 2024. [Online]. Available: https://justenergy.com/blog/power-grid-what-is-it-and-how-does-it-work/

[11] Greenhouse Gas Protocol, "Calculation tools and guidance," https://ghgprotocol.org/calculation-tools-and-guidance, 2023.

[12] 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. [Online]. Available: https://doi-org.proxy-ub.rug.nl/10.1145/3607114.3607117

[13] I. Khan, "Temporal carbon intensity analysis: renewable versus fossil fuel dominated electricity systems," *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 41, no. 3, pp. 309–323, 2019. [Online]. Available: https://doi.org/10.1080/15567036.2018.1516013

[14] S. Meyn, P. Barooah, A. Busic, Y. Chen, and J. Ehren, "Ancillary service to the grid using intelligent deferrable loads," *IEEE Transactions on Automatic Control*, vol. To appear, 12 2014.

[15] M. Bakare, A. Abdulkarim, M. Zeeshan, and A. Nuhu, "A comprehensive overview on demand side energy management towards smart grids: challenges, solutions, and future direction," *Energy Informatics*, vol. 6, 03 2023.

[16] J. Guitart, "Toward sustainable data centers: a comprehensive energy management strategy," *Computing*, vol. 99, no. 6, p. 597–615, jun 2017. [Online]. Available: https://doi.org/10.1007/s00607-016-0501-1

[17] A. Radovanovic, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, S. Talukdar, E. Mullen, K. Smith, M. Cottman, and W. Cirne, "Carbon-aware computing for datacenters," 2021.

[18] S. Tripathi, P. Kumar, P. Gupta, R. Misra, and T. Singh, "Workload shifting based on low carbon intensity periods: A framework for reducing carbon emissions in cloud computing," in *2023 IEEE International Conference on Big Data (BigData)*, 2023, pp. 3387–3395.

[19] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Adv. Sci.*, vol. 8, p. 2100707, 2021.

[20] L. Bouza, A. Bugeau, and L. Lannelongue, "How to estimate carbon footprint when training deep learning models? a guide and review," *Environmental Research Communications*, vol. 5, no. 11, p. 115014, nov 2023. [Online]. Available: https://dx.doi.org/10.1088/2515-7620/acf81b

[21] A. Karyakin and K. Salem, "An analysis of memory power consumption in database systems," in *Proceedings of the 13th International Workshop on Data Management on New Hardware*, ser. DAMON '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi-org.proxy-ub.rug.nl/10.1145/3076113.3076117

[22] TechTarget, "Power usage effectiveness (pue)," https://www.techtarget.com/searchdatacenter/definition/power-usage-effectiveness-PUE, 2023, accessed: 2023-10-05.

[23] Cloud Carbon Footprint, "Cloud carbon footprint documentation," https://www.cloudcarbonfootprint.org/docs/, 2023.

[24] Etsy, "Cloud jewels: Estimating kwh in the cloud," https://www.etsy.com/codeascraft/cloud-jewels-estimating-kwh-in-the-cloud, 2023.

[25] N. Tan, J. Liu, and S. Fan, "Research on the application of deep learning methods in carbon emission prediction," in *2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT)*, 2023, pp. 459–463.

[26] W. d. Assis Pedrobon Ferreira, I. Grout, and A. C. Rodrigues da Silva, "Forecasting energy time-series data using a fuzzy artmap neural network," in *2020 International Conference on Power, Energy and Innovations (ICPEI)*, 2020, pp. 1–4.

[27] M. Richards. (2024) Layered architecture. [Online]. Available: https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html

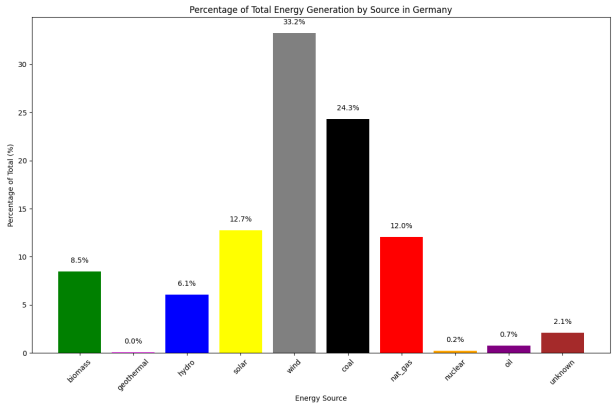
[28] European Association for the Cooperation of Transmission System Operators, "Entsoe transparency platform," Online, 2008, accessed: 2022-05-06. [Online]. Available: https://transparency.entsoe.eu/

[29] Standard Performance Evaluation Corporation. (2024) Specpower_ssj2008 benchmark. Accessed: 26 July 2024. [Online]. Available: https://www.spec.org/power_ssj2008/

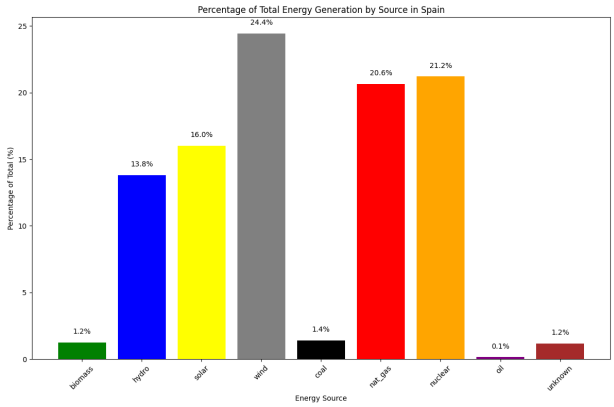
[30] DOC/NOAA/NWS/NCEP, "Ncep gfs 0.25 degree global forecast grids historical archive," https://rda.ucar.edu/datasets/ds084.1/, August 2024, accessed: 2024-07-28.

[31] G. F. Schreiber. (2024) Philadelphia weekend weather & storm forecasts. [Online]. Available: https://theweatherguy.net/blog/

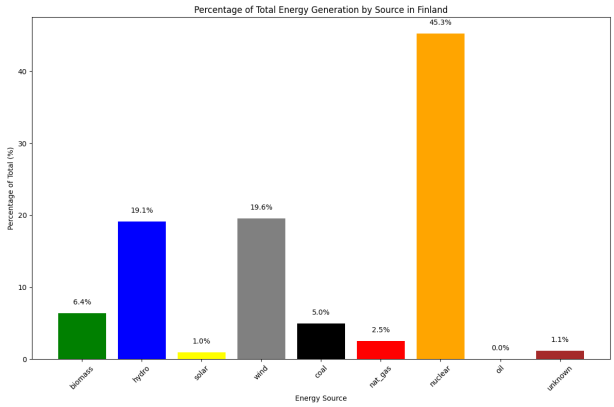
VII. Appendix



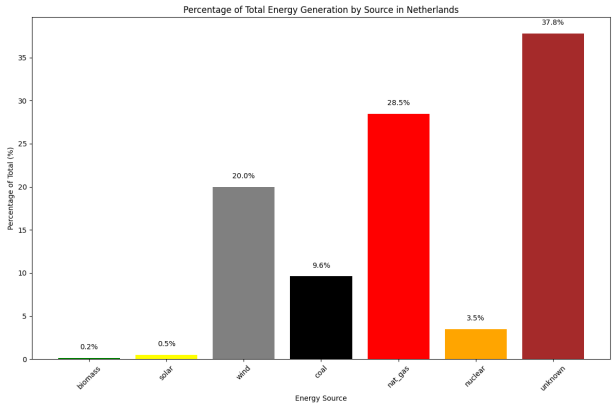
(a) Germany



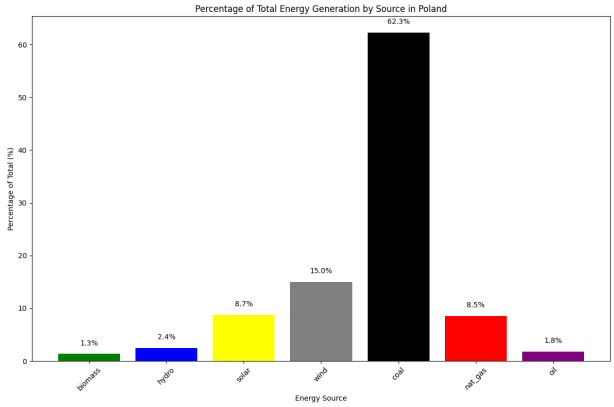
(b) Spain



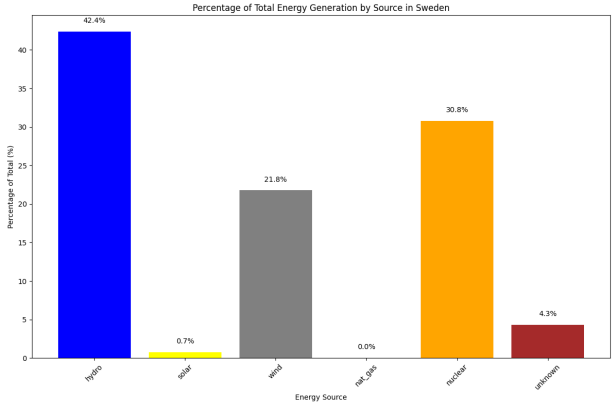
(c) Finland



(d) Netherlands

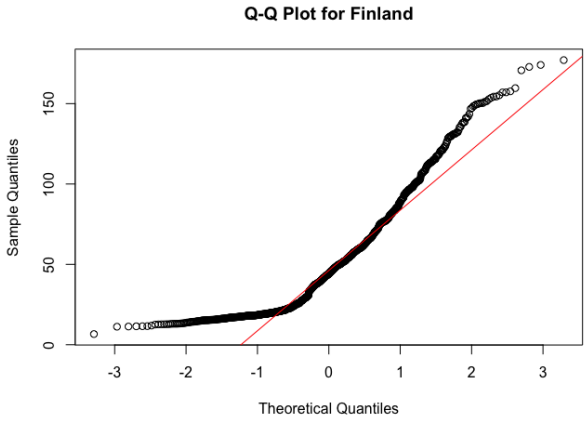


(e) Poland

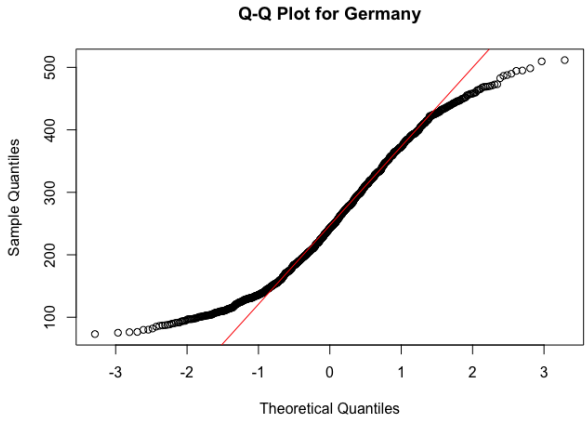


(f) Sweden

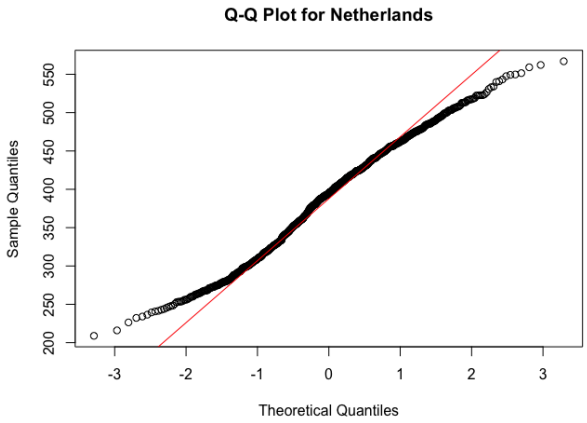
Fig. 10: Percentage of All Energy Sources in Various Countries.



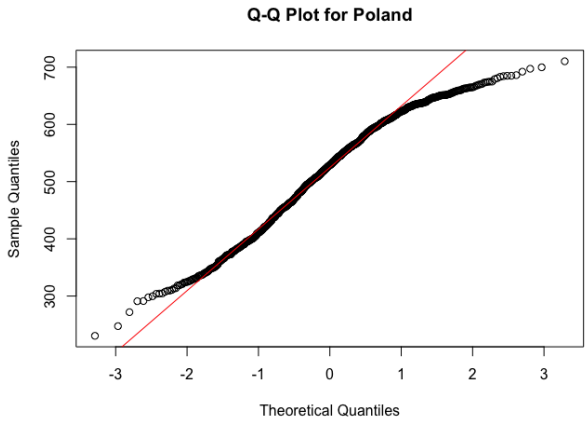
(a) Finland



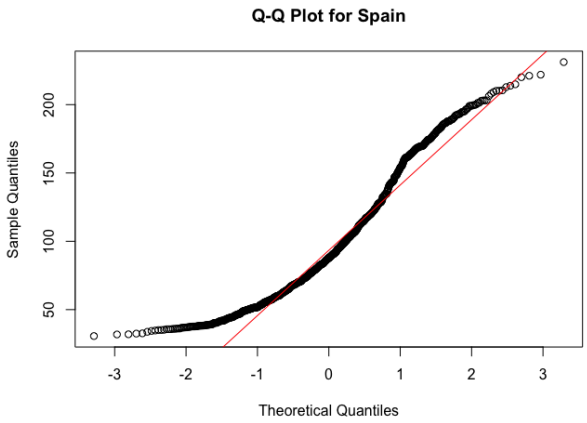
(b) Germany



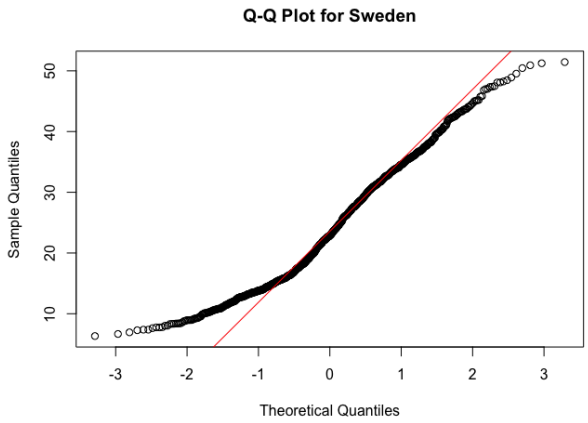
(c) Netherlands



(d) Poland



(e) Spain



(f) Sweden

Fig. 11: Q-Q Plots for assessing normality in various countries.

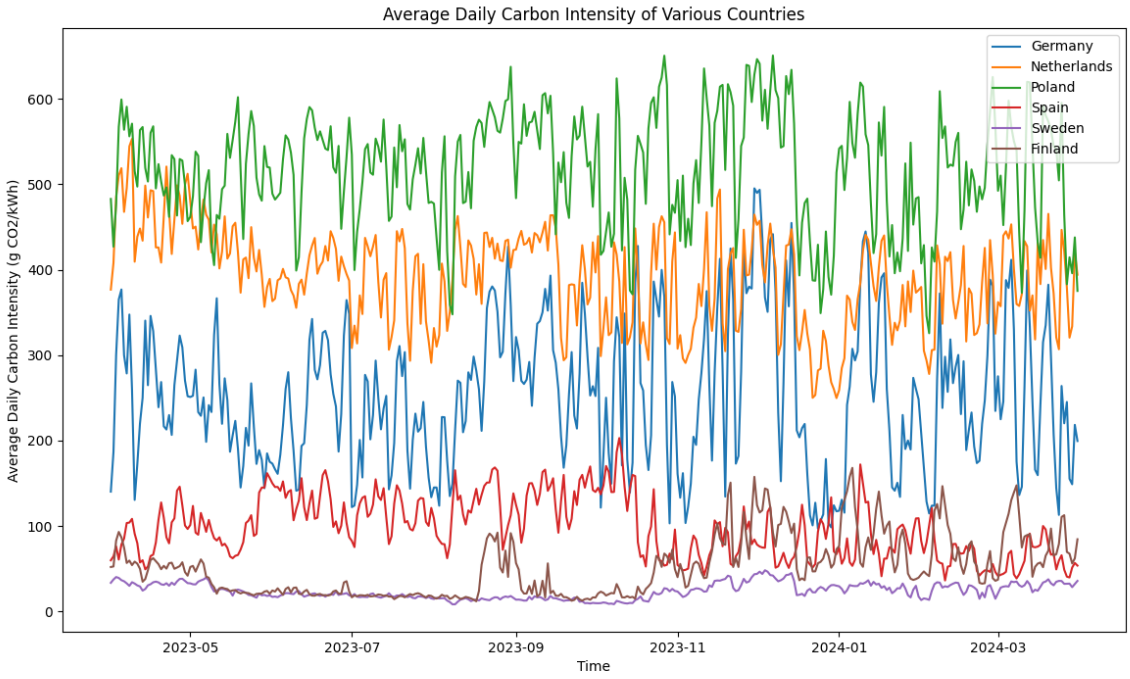


Fig. 12: Average Daily Carbon Intensity of Various Countries from May 2023 to April 2024. This graph illustrates the fluctuations in carbon intensity across different countries. Noticeable deviations can be observed with countries like Poland and Germany exhibiting higher peaks compared to others. Poland, largely dependent on coal, shows the highest carbon intensity, particularly during the colder months, indicating higher coal usage for heating and energy. Germany, with a mixed energy profile, experiences fluctuations that correspond with changes in renewable energy output and coal use. In contrast, Sweden and Finland show significantly lower intensities, highlighting their effective use of renewable resources like hydro and nuclear power. The seasonal variations in carbon intensity reflect changes in energy demand and production efficiency, impacting the carbon footprint associated with electricity consumption.