future internet

Article

Measuring the Effectiveness of Carbon-Aware AI Training Strategies in Cloud Instances: A Confirmation Study

Roberto Vergallo *,†,‡ **and Luca Mainetti** †,‡

Department of Innovation Engineering, University of Salento, 73100 Lecce, Italy; luca.mainetti@unisalento.it
* Correspondence: roberto.vergallo@unisalento.it
† These authors contributed equally to this work.
‡ Current address: Campus Ecotekne, Via per Monteroni 165, 73100 Lecce, Italy.

Abstract: While the massive adoption of Artificial Intelligence (AI) is threatening the environment, new research efforts begin to be employed to measure and mitigate the carbon footprint of both training and inference phases. In this domain, two carbon-aware training strategies have been proposed in the literature: Flexible Start and Pause & Resume. Such strategies—natively Cloud-based—use the time resource to postpone or pause the training algorithm when the carbon intensity reaches a threshold. While such strategies have proved to achieve interesting results on a benchmark of modern models covering Natural Language Processing (NLP) and computer vision applications and a wide range of model sizes (up to 6.1B parameters), it is still unclear whether such results may hold also with different algorithms and in different geographical regions. In this confirmation study, we use the same methodology as the state-of-the-art strategies to recompute the saving in carbon emissions of Flexible Start and Pause & Resume in the Anomaly Detection (AD) domain. Results confirm their effectiveness in two specific conditions, but the percentage reduction behaves differently compared with what is stated in the existing literature.

Keywords: green AI; green software; cloud; sustainability; carbon awareness

check for updates

Citation: Vergallo, R.; Mainetti, L. Measuring the Effectiveness of Carbon-Aware AI Training Strategies in Cloud Instances: A Confirmation Study. *Future Internet* **2024**, *16*, 334. https://doi.org/10.3390/fi16090334

Academic Editor: Gianluigi Ferrari

Received: 7 August 2024
Revised: 11 September 2024
Accepted: 12 September 2024
Published: 13 September 2024

1. Introduction

The environmental impact of technological development, particularly in the realm of software, has become an increasingly relevant and concerning topic in recent years. Technological advancements and the widespread use of software applications and devices have resulted in a range of often underestimated environmental consequences. While digital technologies have undoubtedly enhanced our lives in many ways, they have also led to significant greenhouse gas emissions, increased energy consumption, and waste production.

The rapidly growing sector of the software industry relies on a vast infrastructure, including data centres, servers, network devices, and user devices, all of which require substantial energy to operate [1,2]. In 2018, it was estimated that data centres alone consumed 1% of the global energy supply [3]. The increasing use of Cloud services, the explosion of data generated, and the growing computational power required to run complex algorithms have contributed to a significant rise in energy consumption and carbon emissions [4]. Consequently, improving the energy efficiency of data centres has become a crucial focus for reducing the environmental impact of the software industry [5].

The availability of Cloud instances, providing access to significant hardware resources, has facilitated the widespread use of Artificial Intelligence (AI). In recent years, the use of AI has experienced unprecedented growth, revolutionising various industries and profoundly impacting our daily lives. The substantial computational resources needed to train deep neural networks and run complex algorithms have led to a significant increase in CO_2eq emissions [6]. Recent research has highlighted the environmental impact of AI; for instance, training GPT-2 was found to produce emissions five times greater than those of a car throughout its entire lifecycle [7].

The infrastructure supporting these algorithms, including the energy supply and efficiency of data centres, plays a significant role in the overall calculation of emissions. Furthermore, fully harnessing clean energy is also an ethical issue. Whenever non-renewable energy sources are prioritised, the opportunity to utilise cleaner energy is lost due to inefficient storage systems [8]. Therefore, in this context, the efficient utilisation of the involved technologies becomes relevant. Addressing and mitigating the environmental impact of AI requires a comprehensive approach that considers both the computational resources involved and the efficiency of the infrastructure supporting these technologies.

The purpose of this paper is to confirm the effectiveness of two carbon-aware AI training strategies: Flexible Start and Pause & Resume. As their names suggest, these strategies temporally shift workloads to optimally schedule the training of AI algorithms, aligning high computational loads with the most favourable moments on the grid. Carbon emissions on the grid vary considerably throughout the day and from region to region, due to the intermittent availability of renewable energy sources. These two strategies, which have already been benchmarked in the literature, wait for the optimal time to start or resume the training workload based on a carbon intensity threshold.

The two strategies, which have been previously benchmarked in the literature, aim to initiate or resume training workloads during periods of low carbon intensity. Flexible Start waits for the best moment to start the training workload based on a carbon intensity threshold, while Pause & Resume pauses the training when carbon intensity is high and resumes it when it falls below the threshold.

In a previous study [9], Flexible Start and Pause & Resume were tested on a benchmark of 11 AI workloads in the domains of Natural Language Processing (NLP) and Computer Vision across 16 different geographical regions. The results were encouraging: Flexible Start saved up to 26.6% in emissions for short-duration workloads, while Pause & Resume achieved up to 11.4% carbon savings for the longest workloads, such as a 6.1B parameter language model. Unfortunately, this study is the only one providing quantitative evaluations of these strategies to date. To confirm their validity across various AI tasks and regions and to encourage their adoption in the industry, further research is needed to assess the consistency of these results under different conditions.

The research questions we address in this work are the following:

- RQ1: How effective are the Flexible Start and Pause & Resume carbon-aware AI training strategies in reducing carbon emissions when applied to different Anomaly Detection algorithms?
- RQ2: What impact do geographical variations, specifically the carbon intensity of different regions, have on the carbon savings achieved by the Flexible Start and Pause & Resume strategies?
- RQ3: How do the carbon savings from Flexible Start and Pause & Resume strategies differ when applied to Machine Learning (ML) and Deep Learning (DL) Anomaly Detection algorithms, and what factors influence these variations?

To answer the provided research question, in this work we benchmark Flexible Start and Pause & Resume for four Anomaly Detection (AD) algorithms [10], utilising a large dataset of real financial transactions provided by an Italian bank. These algorithms include a variety of approaches, encompassing both Machine Learning (ML) and Deep Learning (DL). We apply the strategies in the North Carolina region as a reference, following the methodology of other works in the field. Additionally, we average their outcomes across seven European data centres within the AWS network. Energy consumption for the four training algorithms was measured on a DGX machine by NVidia (Santa Clara, CA, USA) equipped with 8 A100 GPUs per node, provided by CINECA (Consorzio Interuniversitario dell'Italia Nord Est per il Calcolo Automatico), a university consortium supporting research activities through the use of supercomputers. Our results confirm the effectiveness of the proposed carbon-aware training strategies, revealing an interesting correlation between the amount of carbon savings, the region in which these strategies are applied, and the time extension assigned to the strategies.

The rest of the paper is structured as follows. In Section 2, we report an analysis of the state-of-the-art. Section 3 summarises the formulation of the two benchmarked strategies. Section 4 presents the research methodology adopted in this paper, also highlighting some technical details on the conducted experiment. Section 5 presents the results of our benchmark. In Section 6, we discuss the main results, comparing them with those already present in the state-of-the-art. Finally, Section 7 outlines the main outcomes of this research and sketches future research challenges.

2. Relevant Literature

With the aim of ensuring a comprehensive and unbiased understanding of the current knowledge on the subject, in this section we explore the current body of literature in the field of Green AI. To do so, we use the results of a recent systematic review [11], which are provided via a replication package https://github.com/luiscruz/slr-green-ai (accessed on 11 September 2024) available online with an open-source license. The study goes through the analysis of 98 peer-reviewed papers, observing a notable surge in research activity since 2020. The review provides a comprehensive overview of the various topics addressed within Green AI literature. Particularly, 13 distinct areas of focus emerge, with primary attention given to monitoring, hyperparameter tuning, model benchmarking, and deployment. Less commonly explored topics, such as data-centric approaches, estimation, and emissions, reveal research gaps that warrant further investigation in the coming years.

In Figure 1, authors provide the distribution of publications across different Green AI definitions, i.e., the level of abstraction used in the paper to quantify the impact of AI in the surrounding environment: energy efficiency [12], carbon footprint [13], or ecological footprint [14]. The figure shows that only 20 papers (∼22% of primary studies) are focused on the topic of the AI carbon footprint, despite, as demonstrated in recent literature [15], reducing the environmental impact of AI exclusively to energy consumption has to be deemed an overly simplistic process.

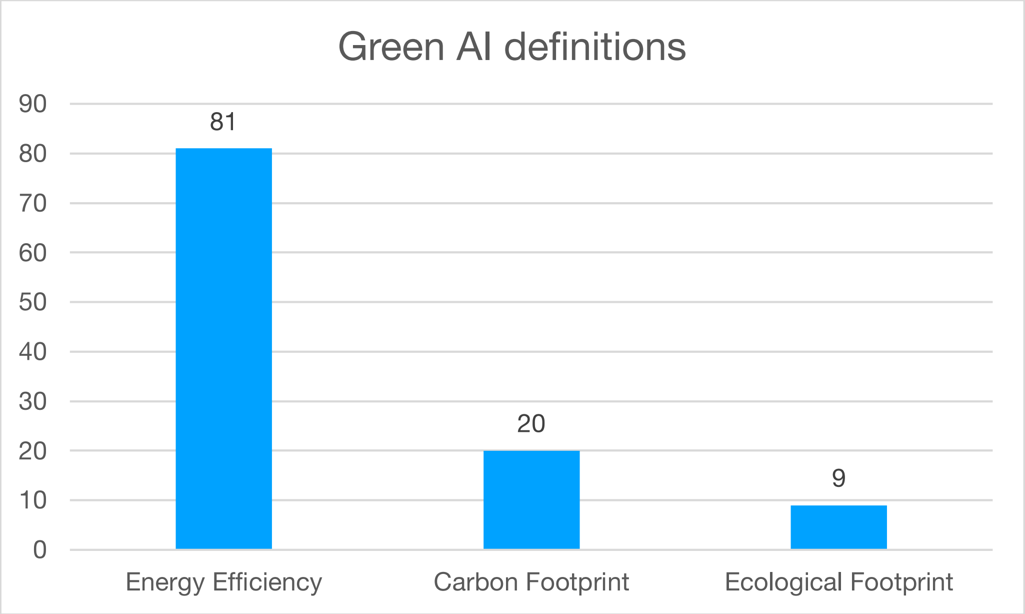


Figure 1. Number of publications per type of Green AI definition (source [11]).

The 20 primary studies are split between observational [7,9,16–21], position [6,22–27], and solution [28–32]. The eight observational papers represent the most relevant literature for this study; hence, we briefly report their contents. Strubell et al. [7] present a seminal paper in which the authors assess the carbon impact of training state-of-the-art models.

Their findings underscore the necessity of reducing the carbon footprint associated with the development and deployment of AI models, sparking significant self-reflection within the AI research community. Dodge et al. [9] provide the first benchmarking of Flexible Start and Pause & Resume training strategies, which will be discussed in detail in the next section. Fraga-Lamas et al. [16] focus on understanding the carbon footprint of creating and using AI systems. Specifically, they examine the energy consumption of an AI-powered IoT scenario and report the corresponding carbon emissions based on different countries and energy sources. Wu et al. [17] explore the energy impact of state-of-the-art models throughout their lifecycle. They offer an overview of the carbon footprint of AI models at Facebook, revealing that approximately 50% of the lifetime carbon cost of AI models stems from the embodied carbon of the hardware used for model development. However, they also highlight that the majority of training workflows significantly underutilize GPUs, operating at just 30–50% of their full capacity. In [18], the authors conduct a series of experiments involving twenty-seven datasets, two decision tree algorithms, and two ensemble methods for classification and regression tasks in order to assess the energy efficiency of these ML algorithms and investigate how design parameters influence energy consumption. Asperti et al. [19] evaluate the energy cost of various variational autoencoders, providing a valuable benchmark for comparing the energy footprint of different models and training techniques. In [20], the authors demonstrate how model distillation techniques can be used to make a machine translation system more sustainable. In a simulated low-resource German-to-English translation task, they show that distilled student models can reduce emissions and monetary costs by nearly 50%. Lastly, Bannour et al. [21] review the tools available to measure the energy consumption and CO_2 emissions of Natural Language Processing (NLP) methods. They compare six tools (carbon tracker, experiment impact tracker, Green algorithms, ML CO_2 impact, energy usage, and cumulator) on named entity recognition experiments across different computational setups (local server vs. computing facility).

3. Carbon-Aware Training Strategies

Dodge et al. [9] propose two carbon-aware training strategies for ML algorithms based on the use of Cloud architectures. As expressed by the authors themselves, they introduce the first tool to estimate the real-time CO_2eq emissions impact of instances on a Cloud computing platform. The previously reported mapping study on Green AI [11] confirmed that their algorithms represent the state of the art of this context. Their study is a post-hoc analysis that investigates what the CO_2eq emissions would have been if a given workload had been performed under different conditions and in different ways. As workloads, they considered various model training tasks in NLP and computer vision. For each training task, they calculated the operational emissions using the following:

- GPUs' energy consumption tracked each 5 min;
- Historical data of local marginal emissions supplied by WattTime with a 5 min granularity.

Particularly the 11 training workloads are split in NLP (BERT Pretraining, BERT Finetuning, Billion Parameter Transformer) and Computer Vision (DenseNets 121, DenseNets 169, DenseNets 201, Vision Transformers Tiny, Vision Transformers Small, Vision Transformers Base, Vision Transformers Large, Vision Transformers Huge).

In the next subsections, we report the details on the two experimented carbon-aware training strategies: Flexible Start and Pause & Resume.

3.1. Flexible Start

Flexible Start is the first strategy proposed. Given a time window, this algorithm aims to find the best starting time in terms of carbon emissions. Considering all possible start times within the time window, it selects the one that would produce the lowest emissions. This approach ensures that the workload runs continuously until completion. More rigorously:

$$O_{\mathrm{fs}} = \min_{i \in [s,e]} (\mathbf{E} \cdot \mathbf{I}_{\mathrm{i}}) \tag{1}$$

where

- $\mathbf{E}$ is the $1 \times n$ row vector of 5 min energy consumption, where n is the number of intervals that compose the workload.
- $\mathbf{I}_{\mathrm{i}}$ is the $n \times 1$ column vector that represents the local marginal emissions at 5 min intervals for the $i - th$ starting time.
- O_{fs} is the operational emissions for Flexible Start strategy.
- i is in the range of s to e, where s is the start-time of the time window, and e is the end time.

3.2. Pause & Resume

This strategy implies that the job could be stopped at certain points during the run and then restarted. Assuming that stopping a job does not require additional costs, the algorithm, within a given time window, selects all the lowest marginal emissions at 5 min intervals until the length of the workload is covered. Once these intervals are determined, the corresponding emissions are computed.

$$O_{\mathrm{p\&r}} = \mathbf{E} \cdot \mathbf{I}_{lowest} \tag{2}$$

where

$$\mathbf{I}_{lowest} = [v_1 | v_2 | ... | v_n] \tag{3}$$

i.e., $\mathbf{I}_{lowest}$ is a $n \times 1$ column vector resulted by the concatenation of the n 5 min lowest intervals of marginal emissions.

3.3. Algorithms Comparison and Application

The implementation of the algorithms is quite simple, but despite this, the results are very interesting. First, the strategies were compared by taking the historical data from 2020 for different regions and using two sets of values for the time window, specifically the following:

- {6, 12, 18, 24} h for Flexible Start.
- {25%, 50%, 75%, 100%} for Pause & Resume.

These values are used to calculate the time window size: calling t the duration of the workload, the first set explains what values need to be added to t to obtain the time window; the second set explains how much larger the time window will be compared with t.

Regarding emissions reduction, the strategies are remarkably efficient in complementary cases.

- Flexible Start. This strategy has been shown to be particularly efficient for short-duration workloads, such as in the case of DenseNet, where the best percentage of emissions reduction reaches 80% in the West US region. The optimal time window was found to be 24 h. In contrast, for jobs longer than one day, the reductions are less significant. For instance, applying this strategy to a 6.1B parameter transformer led to reductions of less than 2%. This may be explained by the fact that shorter jobs are less affected by the variability of marginal emissions during the time window. Therefore, this strategy is applicable to workloads that need to be run regularly but have some flexibility regarding their start times.
- Pause & Resume. This strategy yields considerable reductions in regions with wide variability of marginal emissions throughout the day and for workloads longer than a day. When applied to DenseNet, it brought a small reduction of less than 10%. Conversely, when applied to the 6.1B parameter transformer, the reduction reaches almost 30%.

The authors also conducted an analysis by applying the same time window to the strategies: either an increase of 24 h or an increase of 100% (Table 1).

Table 1. The reductions in percent averaged over the year and across the 16 regions for the Flexible Start and Pause & Resume allowing for a 24 h increase and a 100% increase in job duration. Hours and energy consumption refer to training workload without strategies.

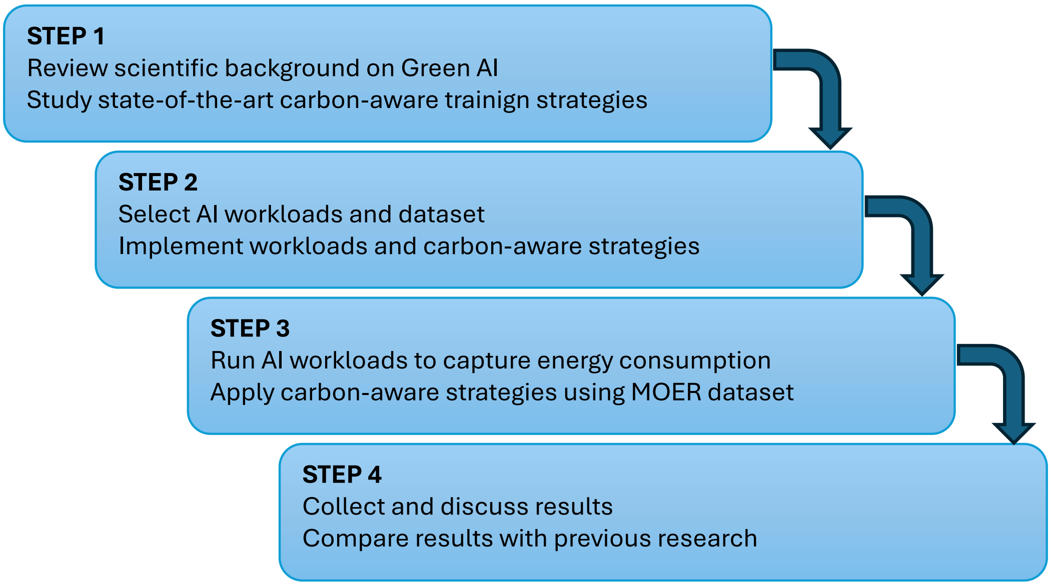
Model		**BERT Fine-tune**	**BERT LM**	**6.1B Trans-former**	**Dense 121**	**Dense 169**	**Dense 201**	**ViT Tiny**	**ViT Small**	**ViT Base**	**ViT Large**	**ViT Huge**
Hours		6	36	192	0.3	0.3	0.4	19	19	21	90	216
kWh		3.1	37.3	13,812.4	0.02	0.03	0.04	1.3	2.2	4.7	93.3	237.6
24 h increase	Flexible Start	14.5%	3.4%	0.5%	26.8%	26.4%	25.9%	5.6%	5.3%	4.2%	1.3%	0.5%
	Pause & Resume	19.0%	8.5%	2.5%	27.7%	27.3%	27.1%	12.5%	12.3%	11.7%	4.7%	2.4%
100% increase	Flexible Start	7.0%	4.1%	2.6%	1.8%	2.5%	2.7%	5.0%	4.8%	3.9%	3.3%	3.0%
	Pause & Resume	9.5%	11.0%	11.4%	2.0%	2.8%	3.1%	11.0%	11.0%	10.8%	11.4%	11.3%

It should be kept in mind that using historical data leads to the best results, so the emission reductions turn out to be lower bounds for a realistic scenario. WattTime also provides forecast data. Considering that Flexible Start does not extend too much over time, these forecasts can be used to achieve realistic results. Regarding Pause & Resume, it is preferable to stop the training whenever emissions exceed a certain threshold and restart when they return to lower levels. The threshold would be set in such a way as to arbitrarily increase the time window. In any case, as of today, even 24 h forecasts are accurate, so the application through their use would yield reliable results.

4. Materials and Methods

4.1. Research Methodology

In Figure 2, we report the adopted research methodology. It is made of four steps, enumerated from 1 to 4.



STEP 1
Review scientific background on Green AI
Study state-of-the-art carbon-aware trainign strategies

STEP 2
Select AI workloads and dataset
Implement workloads and carbon-aware strategies

STEP 3
Run AI workloads to capture energy consumption
Apply carbon-aware strategies using MOER dataset

STEP 4
Collect and discuss results
Compare results with previous research

Figure 2. Research methodology adopted in this paper.

Step 1 consists of what we have already done in the previous sections, i.e., an overall review of the scientific background on the topic of Green AI and a careful study of the state-of-the-art carbon-aware strategies, namely, Flexible Start and Pause & Resume.

Once these strategies have been identified and analysed, the idea is to re-implement them and conduct benchmarks to confirm their validity in different conditions by performing an in vitro experimentation. To do this, it is necessary to choose different workloads

(Step 2). Specifically, in this work, we selected algorithms from the field of AD, particularly fraud detection, because of the availability of a large dataset of real banking transactions provided by a well-known Italian bank.

Therefore, we tracked the energy consumption of training those AD algorithms, and then we projected such energy needs on a dataset of historical Marginal Emissions Operating Emissions Rate (MOER) provided by the research partner WattTime (Step 3). The idea is to conduct a retrospective analysis to understand what the carbon emissions could be if training were executed in one way rather than another. These training jobs were executed on a remote machine, and their energy consumption was monitored during the execution.

Once the energy consumption data is obtained and mapped with the marginal emissions, benchmark calculations were performed to provide a comparative analysis of the different strategies (Step 4). Specifically, all strategies were executed under the same conditions for the considered year, 2021, as well as regions and days for each month. The idea is to calculate the percentage reductions in carbon emissions for each strategy by averaging the results over the entire year of 2021. For each month, we selected six equal starting times, corresponding to six days. This choice is made to accurately describe the average emissions for each month. This approach aligns with the methodology used by Dodge et al. [9] to summarise annual emissions.

4.2. Anomaly Detection AI Algorithms

The domain of AI extends to the field of Anomaly Detection (AD), encompassing a variety of approaches. Specifically, it ranges from unsupervised to supervised learning and even includes DL algorithms. In this subsection, we provide a brief review of the most common approaches. As mentioned in the previous section, this work focuses on the case study of fraud detection, which is a category within AD. The choice of this domain is driven by two specific reasons:

1. Environmental impact: While there are AI sectors with more significant emissions, it is important to also pay attention to other sectors, such as AD, which will be demonstrated to have a non-negligible environmental impact.
2. Data accessibility: This study uses a dataset provided by an Italian bank containing both fraudulent and non-fraudulent banking transactions. Access to a real dataset, rather than a synthetic one, allows for more realistic evaluations and assessments.

As mentioned earlier, this case study employs supervised learning algorithms. In supervised learning, labelled datasets are used, where each record is labelled as either a fraudulent or non-fraudulent transaction. In contrast, unsupervised learning uses unlabelled records and allows them to form clusters autonomously.

- Isolation Forest (IF) [33]. This algorithm is used in AD to identify anomalies within a dataset. It is based on an unsupervised learning approach and operates on the principle that anomalies are rarer and therefore more "isolated" than normal instances. The algorithm works by creating a set of random decision trees, known as isolation trees. Each tree is constructed by randomly selecting a subset of the data and recursively partitioning it based on random attribute values. The splitting process continues until each data instance is isolated in a single terminal node of the tree. To identify anomalies, the algorithm evaluates the separation path within the trees. Instances that require fewer splits to be isolated are considered more anomalous. Therefore, anomalies will have a shorter average path length compared with normal instances. The Isolation Forest has several advantageous features, including its ability to handle large-scale datasets efficiently, robustness to variations in data dimensionality, and capability to detect different types of anomalies, such as point anomalies and collective anomalies. However, it is important to note that the Isolation Forest may not be effective in detecting certain complex or subtle anomalies that require more contextual analysis. Thus, it is often used in combination with other AD algorithms to improve overall system performance.

- Support Vector Machine (SVM) [34]. This is a ML algorithm used for classification and regression problems. It is a supervised learning model that can separate data examples into different classes. The basic idea of an SVM is to find the optimal hyperplane that divides the feature space to maximise the separation between different data classes. The hyperplane is selected to be the maximum distance from a set of training examples of one class (called support vectors) while maximising the distance from the support vectors of the other classes. In practice, an SVM uses a set of labelled training examples to train a model that can subsequently classify new unlabelled examples. During the training phase, the SVM algorithm optimises an objective function that balances the maximum separation between classes with the reduction of classification error. The use of kernel functions is an important aspect of SVMs. A kernel allows data to be mapped into a high-dimensional feature space, where it is easier to find a linearly separating hyperplane. This enables SVMs to address nonlinear classification problems. SVMs have proven effective in a variety of applications, including image recognition, text analysis, bioinformatics, and more. They are known for their ability to handle large datasets and generalise well even on unseen data during training.
- HF-SCA [35]. This algorithm is a U-Net, a specific neural network whose architecture resembles the letter "U". This structure is designed such that information, starting from the top left of the "U", is progressively compressed through downsampling using convolutional layers and pooling. Once it reaches the bottom point, the information travels back up towards the top right of the "U", reconstructing the data through upsampling. In the case of HF-SCA, depending on the reconstruction error of the input data, the algorithm determines whether a transaction is fraudulent or not. Additionally, the uniqueness of this algorithm lies in replacing the convolutional layers of the U-Net with squeeze-and-excitation blocks, which have improved performance for the specific case being examined.
- Autoencoder (AE) [36]. This is a neural network architecture commonly used in AD. It falls under the category of unsupervised learning, as it learns to reconstruct the input data without relying on labelled examples. The main idea behind the autoencoder is to use an encoder network to compress the input data into a lower-dimensional representation, also known as the latent space. This compressed representation contains the most essential features of the input data. Then, a decoder network reconstructs the original input data from the latent space representation. During the training phase, the autoencoder aims to minimise the reconstruction error, which is the difference between the original input data and the reconstructed output. By learning to reconstruct normal instances accurately, the autoencoder becomes less capable of reconstructing anomalous instances, leading to higher reconstruction errors for anomalies.

The dataset for training the four AI workloads was provided by a major Italian bank as part of the Sandbox project [37] by Bank of Italy. It consists of a large set of banking transactions, specifically time series data where each transaction is ordered chronologically. It is a tabular dataset with 32 features (columns), including information like customer ID, transaction time, time elapsed from the first transaction in the dataset, merchant name, merchant category code, merchant city and country, and transaction amount. Each row contains information on the specific transaction and also the same information for the last physical transaction (i.e., in store) for that customer, needed for geolocalization. The dataset is labelled, meaning that in addition to the 32 available features, it includes an additional feature indicating whether a transaction is fraudulent. Sensitive features are anonymized, while all the features are scaled through both standard and min-max scalers. The number of rows is 4.5 million. The size of the dataset is 2 GB.

For each algorithm, a grid search was performed, which extended over several hours depending on the training. We chose the grid search approach to simulate the worst-case scenario for tracking energy consumption. The grid search was based on the AUC (Area Under the Curve) score metric, which is the preferred metric for AD due to its robustness in the context of imbalanced datasets. This is particularly relevant in this case study, where the number of anomalies is significantly smaller compared with the rest of the data.

4.3. Benchmarking Software

From a technical perspective and at a lower level, the performances of the strategies were calculated using two Python scripts. It was decided to split the software into two projects:

- Workload Runner. The runner script is deployed on a remote machine with SSH access provided by CINECA, the largest computing centre in Italy. The purpose of the script is to execute ML and DL trainings while monitoring their energy consumption. Specifically, this runner script handles four different workloads and launches the selected workload. Each workload trains its corresponding model using a dataset in .csv format and, upon completion, generates an additional .csv file containing the energy consumption data for that workload. This file is required for the execution of the subsequent script.
- Strategy Launcher. The launcher script is executed on the local machine and simulates the application of one of the carbon-aware training strategies to the chosen workload. This simulation uses the .csv file of energy consumption produced by the Workload Runner script. Once this file is transferred to the local machine via SCP, the launcher script calculates the CO_2eq emissions by mapping the obtained data against marginal emissions for a specific region and day in the year 2021. These marginal emissions data are also obtained from .csv datasets provided by WattTime. After calculating the emissions, the script computes the percentage reductions and visually presents a comparative analysis of the results for all the carbon-aware strategies.

Python version 3.9 was chosen as the programming language for its elegance in syntax and its extensive library support, particularly in the field of AI algorithm development. Python is widely used in the AI community.

Several significant libraries were employed in this project, including the following:

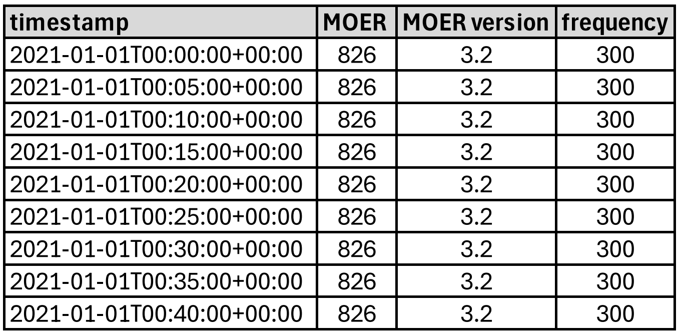
- NumPy [38]: a library for scientific computing used for scalar product calculations, as well as determining maximum, minimum, and average values.
- Pandas [39]: a library for easy handling of structured data such as CSV files. It was primarily used for reading datasets and performing preprocessing tasks, including handling incomplete or dirty data.
- scikit-learn, TensorFlow, and PyTorch [40–42]: well-known libraries that implement ML and DL algorithms. These libraries provided the necessary functions for implementing the four workloads, including metric functions and data preprocessing.
- CodeCarbon [43]: The previously mentioned library used for tracking energy consumption. It even offers the ability to directly calculate carbon emissions for a given function using APIs. In this work, we used CodeCarbon only to acquire energy consumption, while emissions were calculated separately using carbon-aware strategies.
- Matplotlib [44]: a library capable of generating static, interactive, and animated graphs in Python. It was used for generating various visualisations, including timesheets for strategies and bar plots for annual percentage reductions.

These libraries greatly contributed to the implementation and analysis of the project, providing powerful tools for data manipulation, algorithm development, energy tracking, and visualisation.

4.4. Computing Environment

The execution of the four workloads was performed on a DGX A100 machine equipped with 8 × Nvidia A100 GPUs per node, provided by CINECA (Consorzio Interuniversitario dell'Italia Nord Est per il Calcolo Automatico), a university consortium that supports research activities through the use of supercomputers. Specifically, these computational resources were made available through ISCRA (Italian SuperComputing Resource Allocation), a project within CINECA that allocates access to their machines based on different classes tailored to user needs. The trainings were launched as jobs through the SLURM scheduler, using batch mode. This non-interactive mode allows for the use of GPUs and all requested resources, with jobs starting once the necessary resources become available. During execution, energy consumption of CPU, GPU, and RAM was tracked using the CodeCarbon library. Specifically, energy consumption in kWh was recorded every 5 min. These trainings were conducted on this machine to take advantage of the computational power of the GPUs.

One of the advantages of the CodeCarbon library is its ability to track energy consumption specific to the designated process rather than the entire machine. This is important because the CINECA machine supports multiple users running jobs simultaneously, which could otherwise distort the estimated power consumption. Since this work focuses solely on operational emissions, the power consumption data obtained was mapped against historical data of local marginal emissions. In this context, our organisation is a general member of the Green Software Foundation and has established a research partnership with WattTime, which provided historical data (Figure 3) for various regions dating back to 2021. For this work, results were evaluated for seven regions in Europe: Italy, Germany, France, the United Kingdom, Ireland, Spain, and Sweden. Additionally, Northern Carolina was used as a reference region for non-region-specific runs. WattTime provides data for this region for preview purposes, and several experiments available online use Northern Carolina as a testbed (e.g., [45–48]).



timestamp	MOER	MOER version	frequency
2021-01-01T00:00:00+00:00	826	3.2	300
2021-01-01T00:05:00+00:00	826	3.2	300
2021-01-01T00:10:00+00:00	826	3.2	300
2021-01-01T00:15:00+00:00	826	3.2	300
2021-01-01T00:20:00+00:00	826	3.2	300
2021-01-01T00:25:00+00:00	826	3.2	300
2021-01-01T00:30:00+00:00	826	3.2	300
2021-01-01T00:35:00+00:00	826	3.2	300
2021-01-01T00:40:00+00:00	826	3.2	300

Figure 3. An example of a dataset provided by WattTime. In this case, it is the .csv file for the Italian region.

The selection of the European regions is due to three main reasons:

- Compliance with privacy policies: The General Data Protection Regulation (GDPR) is a regulation of the European Union regarding the processing of personal data and privacy. It stipulates that personal data can only be transferred to countries or organisations that provide an adequate level of data protection.
- Access to regions that use both renewable and non-renewable energy sources: this allows for a diverse analysis of emissions across different energy profiles.
- Availability of specific cities from the list of AWS Cloud instances: these regions provide a credible testbed for experiments.

Regarding the datasets, they consist of .csv files with several features, each containing marginal emission information for a specific region over a month. The features include the following:

1. Timestamp: This represents the date and time of each collected sample. As previously mentioned, the data was recorded with a granularity of 5 min.
2. MOER: Marginal Operating Emission Rate, which refers to the rate of carbon emissions emitted per unit of consumed energy, expressed in lbs/kWh. This value has been converted because the power consumption was tracked in kWh, and the carbon emissions were provided in gCO_2eq (grammes of carbon dioxide equivalent).

3. MOER version: refers to the version of the provided MOER data.
4. Frequency: the duration in seconds for which the data is valid from point_time, where point_time is the date/time format indicating when this data became valid.

5. Results

5.1. Baseline

Before reporting the results in terms of carbon emissions savings, we present some baseline metrics about the benchmarked algorithms (Table 2): the AUC score, the energy consumption of the training jobs, and the training time. These metrics are reported for a specific reason: since this work is situated within the context of Green software, the goal is to provide an overview in terms of consumption and accuracy to guide decision-makers in making an informed choice regarding the training process. The aim is to find a balance between model accuracy and its environmental impact. It is acknowledged that HF-SCA stands out significantly among the different workloads. However, as explained earlier, the intention is to consider multiple diverse workloads and analyse their performance and consumption to provide the necessary information for decision-making. This type of proactive decision-making already mitigates emissions, which will be further reduced by the strategies benchmarked in this work.

Table 2. Some basic metrics on the considered AD workloads that can guide the decision makers in a proactive behaviour towards a greener choice.

	Isolation Forest	SVM	HF-SCA	Autoencoder
AUC score	0.56	0.51	0.97	0.73
Energy consumption (kWh)	0.825	0.493	3.310	0.615
Training time (h)	4:15	2:30	16:00	3:30

Additionally, before presenting the results, we provide an overview of the emissions produced by these algorithms to establish a baseline. Energy consumption data was mapped with historical data on local marginal emissions to calculate the average total emissions for the entire year of 2021. This analysis was applied to the seven regions mentioned earlier (Figure 4).

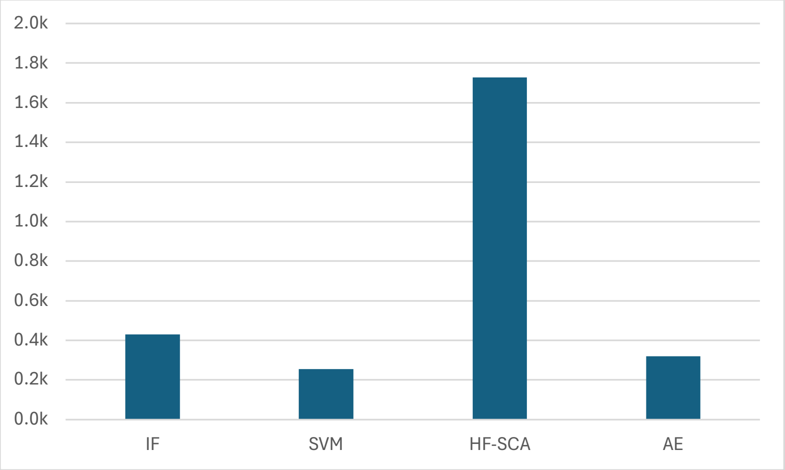


Figure 4. Average emissions of the trainings during the year 2021 across different regions. These emissions exceed 1.6 kg CO_2eq, comparable to emissions of CO_2 per litre of fuel consumed by a car.

Figure 5 shows the carbon emissions of the most polluting workload—the HF-SCA training job—across the entire 2021 year and on the different considered regions. We observe that, except for Ireland, the emission rate remains quite stable throughout the year for all the regions. Additionally, Spain, France, Italy, and the United Kingdom have very similar MOER rates. In contrast, Germany and Sweden exhibit higher emissions, which is particularly surprising for Sweden, given its significant reliance on renewable sources.

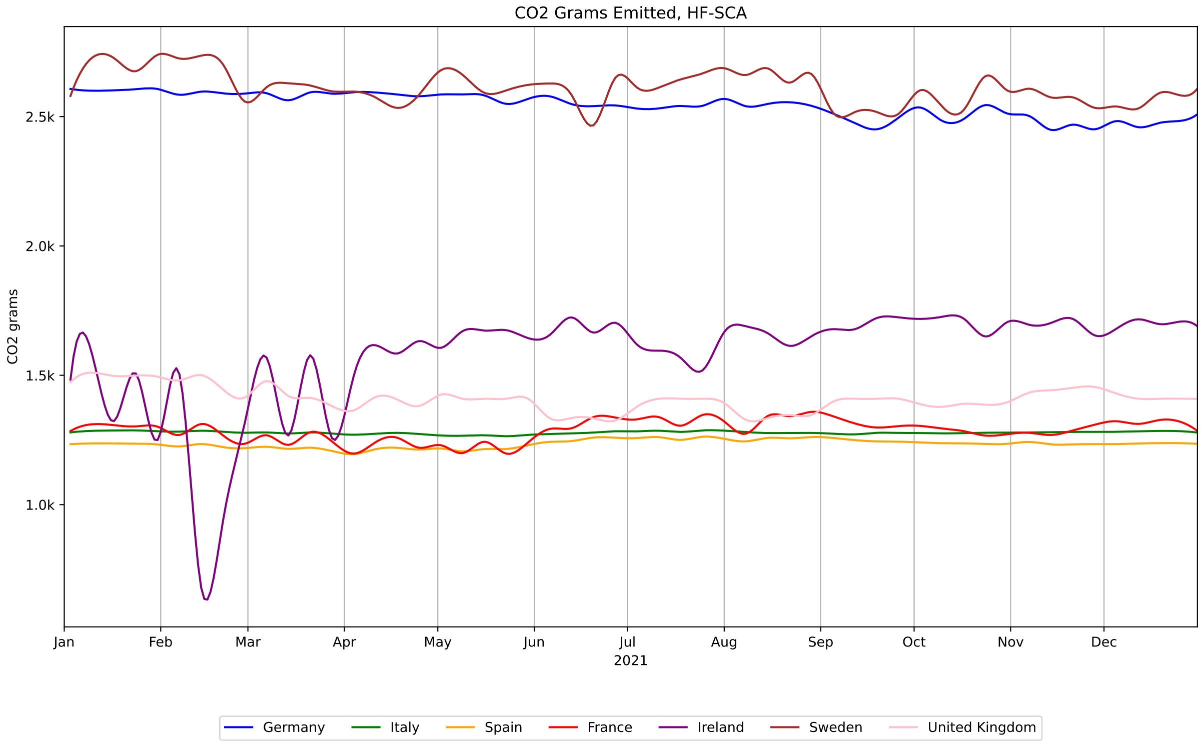


Figure 5. The carbon emissions from training HF-SCA (AD on 8 × A100 GPUs for 16 h) in seven different regions (one region per line) at various times throughout the year are shown. Each line is relatively flat, indicating that emissions in a single region are consistent across different months. However, there is significant variation between the least carbon-intensive regions (represented by the lowest lines) and the most carbon-intensive regions (represented by the top lines). This confirms that selecting the region in which experiments are conducted can have a substantial impact on emissions, with differences ranging from 0.25 kg in the most efficient regions to 2.5 kg in the least efficient regions.

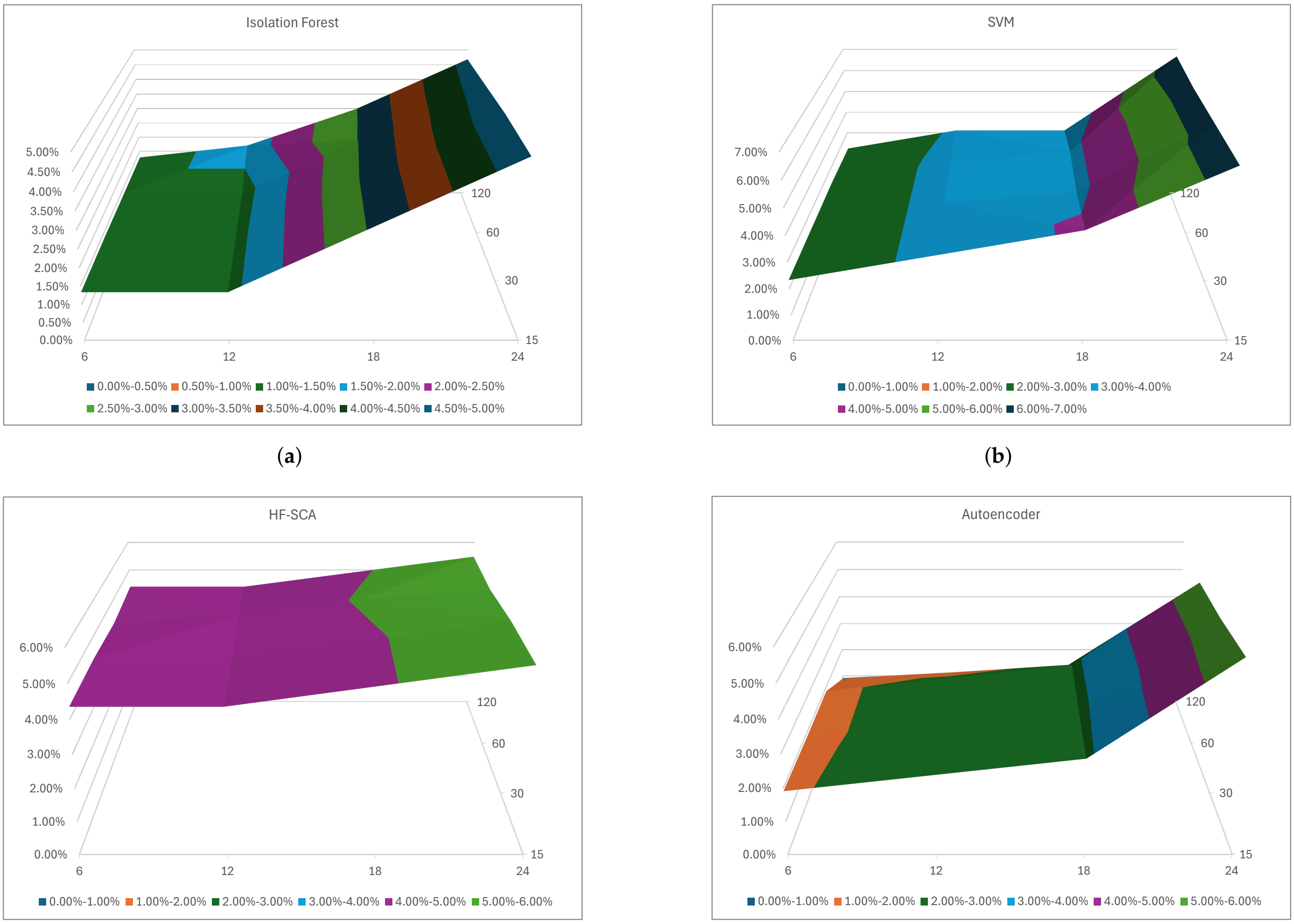
5.2. Emission Reduction

This analysis aims to understand the average reductions in carbon emissions throughout the year 2021 for the two state-of-the-art strategies: Flexible Start and Pause & Resume. For each month, six days are considered for the analysis. This approach has been adopted to closely align with the modus operandi of the authors in [9] and to ensure a reliable comparison of all strategies.

Furthermore, this evaluation involves calculating the reductions based on the variation of key input parameters of the strategies, particularly the following:

- The time window will be calculated by adding each value from the set $[6, 12, 18, 24]$, which we will refer to as the hours-set, to the length of the workload.
- The size of the time window will be calculated by increasing the workload length by each percentage value in the set $[25\%, 50\%, 75\%, 100\%]$, which we will refer to as the percentage set for simplicity.
- The checking time will be set to values in the set $[15, 30, 60, 120]$.

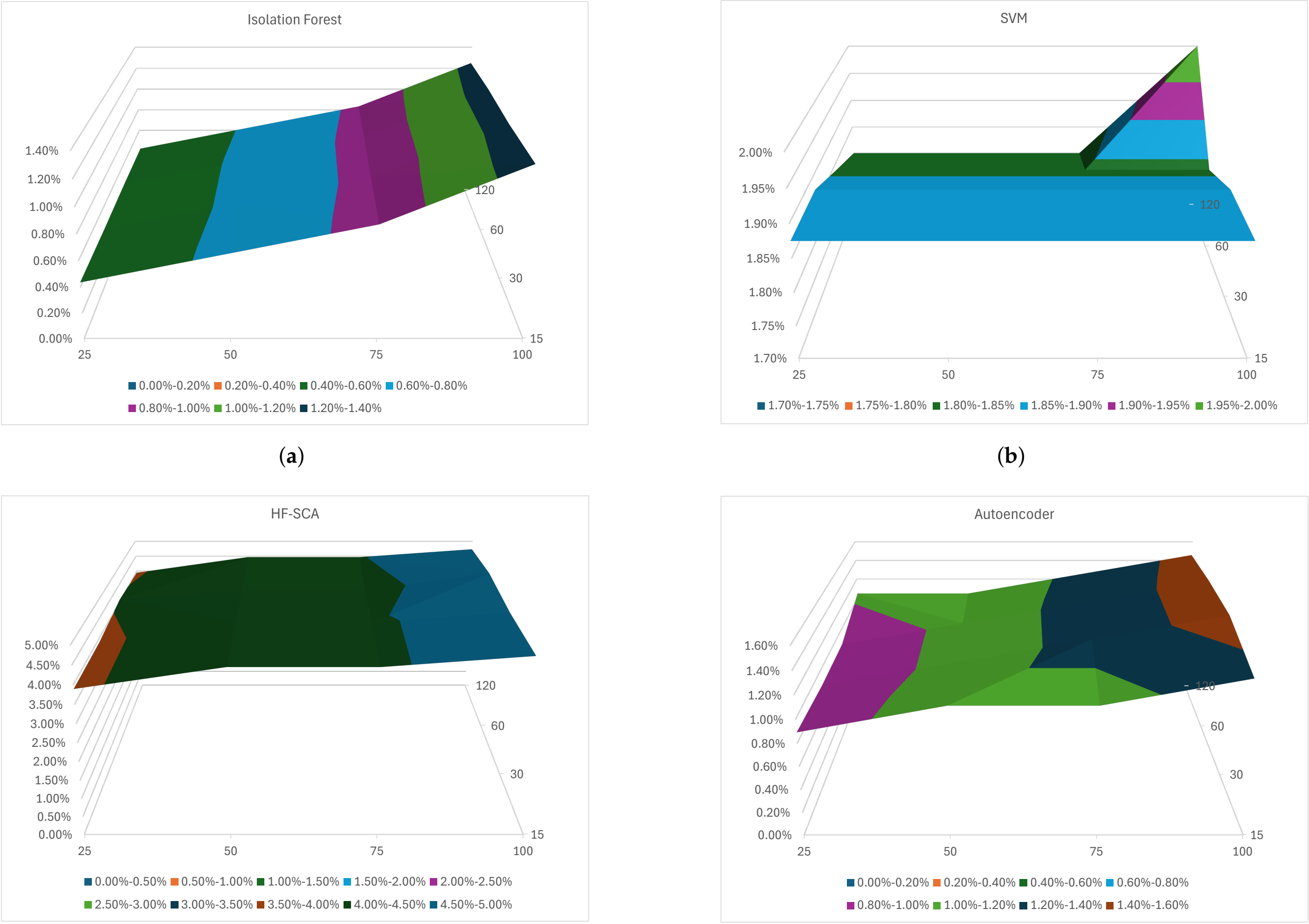
In the next subsections, we conduct a careful analysis for each AI workload, followed by a detailed analysis to offer insights into a real-world scenario. Specifically, we consider the Northern Carolina region as a benchmark. Graphs in Figures 6–9 summarise the obtained results. Graphs are grouped per combination of strategy (Flexible Start/Pause & Resume) and window sizes (percentage/hours set). On the x axis, we have the different percentages in the case of the percentage set (Figures 7 and 9), and the different time extensions in the case of the hours set (Figures 6 and 8). The y axis is the checking time (in minutes). The z axis is the emission reduction percentage. The graphs have colour bands needed to appreciate the height of the surfaces in the 3D space.



(**a**) (**b**)

(**c**) (**d**)

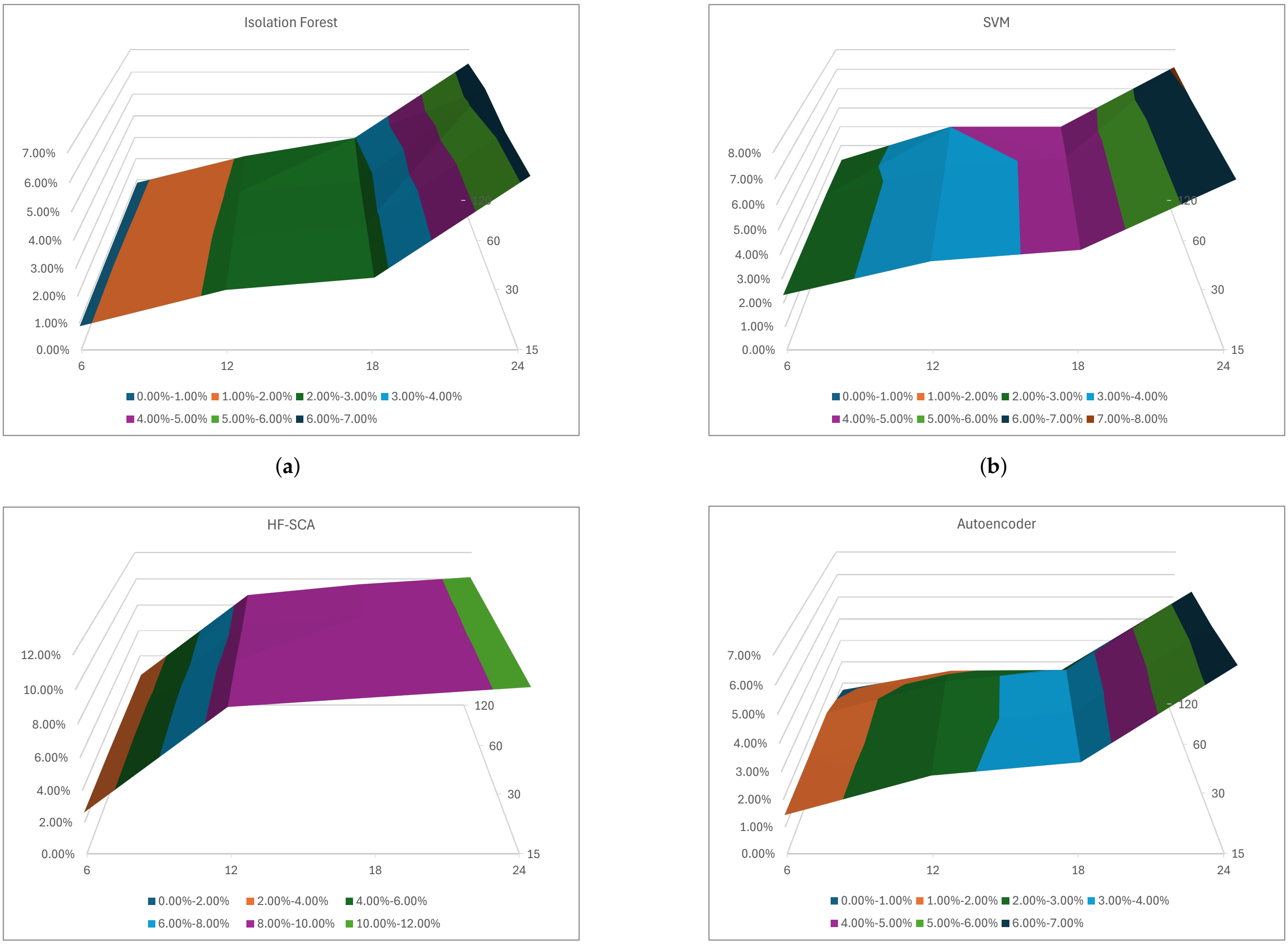
Figure 6. Emission reduction percentage for the four AI workloads using Flexible Start strategy and the hours set in Northern Carolina. The x axis represents the time extension (6, 12, 18, 24 h) assigned to the workload to complete the job. The y axis represents the checking time (15, 30, 60, 120 min) for carbon intensity. The z axis represents the emission reduction percentages for each specific combination of time extension and checking time. (**a**) Isolation Forest workload; (**b**) SVM workload; (**c**) HF-SCA workload; (**d**) autoencoder workload.



(**a**) (**b**)

(**c**) (**d**)

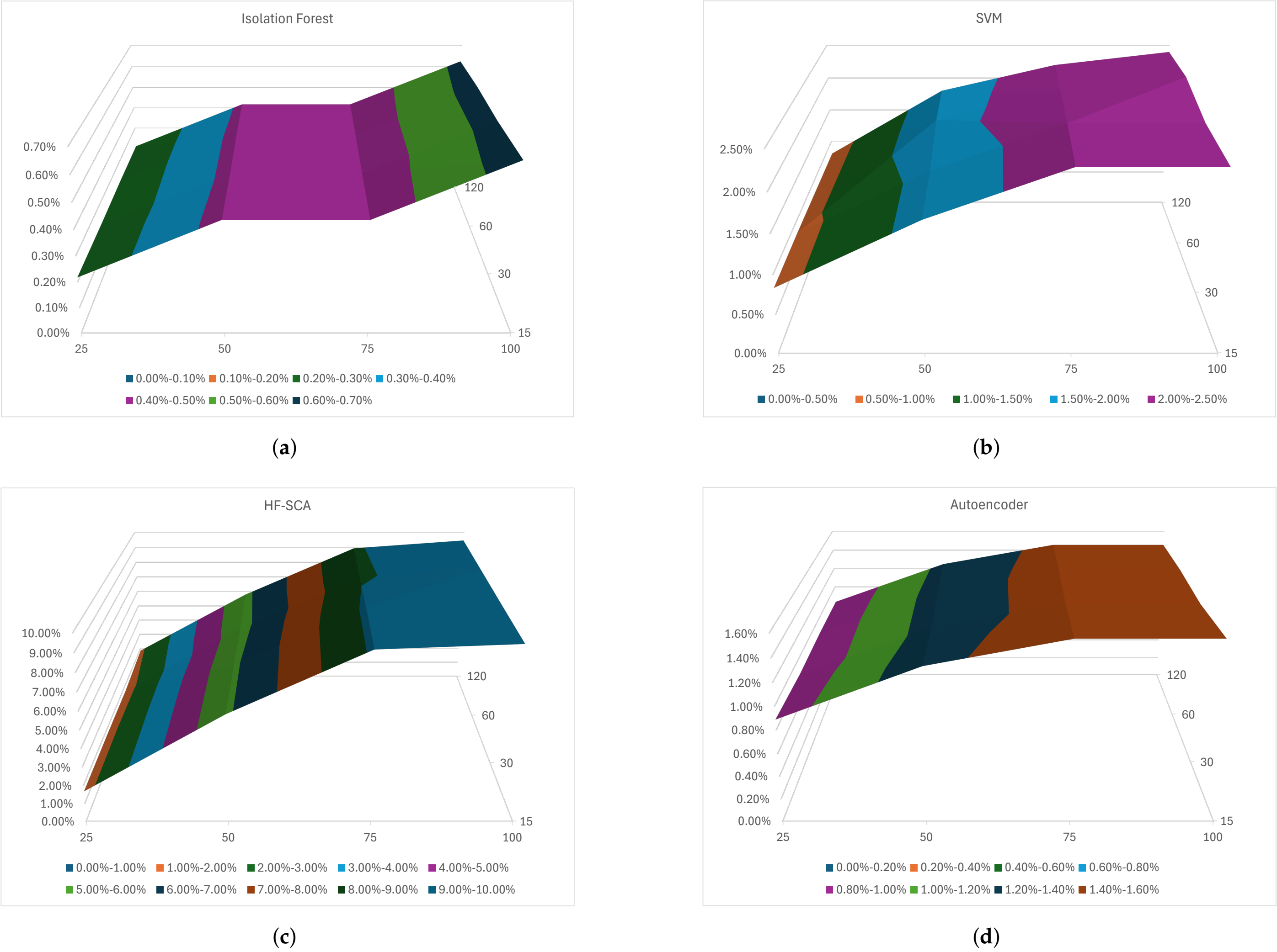
Figure 7. Emission reduction percentage for the four AI workloads using Flexible Start strategy and the percentage set in Northern Carolina. The x axis represents the time extension (+25%, 50%, 75%, 100% of the original training time) assigned to the workload to complete the job. The y axis represents the checking time (15, 30, 60, 120 min) for carbon intensity. The z axis represents the emission reduction percentages for each specific combination of time extension and checking time: (**a**) Isolation Forest workload; (**b**) SVM workload; (**c**) HF-SCA workload; (**d**) autoencoder workload.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Emission reduction percentage for the four AI workloads using Pause & Resume strategy and the hours set in Northern Carolina. The x axis represents the time extension (6, 12, 18, 24 h) assigned to the workload to complete the job. The y axis represents the checking time (15, 30, 60, 120 min) for carbon intensity. The z axis represents the emission reduction percentages for each specific combination of time extension and checking time: (**a**) Isolation Forest workload; (**b**) SVM workload; (**c**) HF-SCA workload; (**d**) autoencoder workload.



(**a**) (**b**)

(**c**) (**d**)

Figure 9. Emission reduction percentage for the four AI workloads using Pause & Resume strategy and the percentage set in Northern Carolina. The x axis represents the time extension (+25%, 50%, 75%, 100% of the original training time) assigned to the workload to complete the job. The y axis represents the checking time (15, 30, 60, 120 min) for carbon intensity. The z axis represents the emission reduction percentages for each specific combination of time extension and checking time: (**a**) Isolation Forest workload; (**b**) SVM workload; (**c**) HF-SCA workload; (**d**) autoencoder workload.

5.2.1. Isolation Forest

Starting with the hours-set (Figure 6a for Flexible Start and Figure 8a for Pause & Resume), it is evident that both strategies tend to increase the percentage of reductions as the value of the hours-set increases. This increment happens because extending the time window by one day compared with the minimum duration of the workload allows for choosing the optimal starting time (as observed by the authors in [9], carbon intensity tends to vary throughout the day).

The checking time with a value of 15 min leads to better results with Flexible Start (Figure 6a). This can be explained by the fact that a shorter check period makes the workload much more sensitive to the variability of the marginal emissions in different regions. With Pause & Resume, however, the best checking time is 60 min (Figure 8a—note the small peak on the blue bar's upper border), which represents a compromise between the bounds of the set (15 and 120). While checking with a lower frequency may lead to missing the MOER variability, checking too often may result in choosing resume moments that are local optima, thus missing better subsequent moments. In general, the maximum average percentage reduction is 6.52% with a value of 24 h for the hours-set. This value is achieved by the Pause & Resume strategy.

In the case of the percentage set (Figure 7a for Flexible Start and Figure 9a for Pause & Resume), regardless of the value taken from it, both strategies show very low percentage reductions. This can be attributed to the duration of the workload. For example, the fact that the Isolation Forest training lasts 4 h implies a temporal window of 8 h if the percentage-set value is 100%. In fact, the reductions are very similar to those calculated based on an hours set value of 6 h (Figure 8a).

5.2.2. SVM

It is the shortest workload among the four considered. Regarding the hours set (Figure 6b for Flexible Start and Figure 8b for Pause & Resume), the average carbon reduction percentage is slightly higher than that of the Isolation Forest. This is remarkable because the reported strategies tend to show better performance on a shorter job (SVM lasts half as long as IF, see Table 2). Both strategies tend to show very similar emission reductions, with the best value of 7.11% achieved by Pause & Resume with a checking time of 120 min. and a 24 h time window.

Regarding the percentage set, both Figures 7b and 9b show remarkable trends. In both cases, the reduction percentage is low (2.00% and 2.45% maximum, respectively), but while Pause & Resume shows a nice plateau, Flexible Start has a notable peak. This should not mislead the reader, as the z axis has a very strict range with a 0.05% unit.

5.2.3. HF-SCA

The HF-SCA workload is the longest among all. Due to its extensive duration, it also results in the highest energy consumption. This duration is also reflected in the achieved AUC score, which is the best according to Table 2. Therefore, this training can emit a significant amount of CO_2, comparable to carbon emissions per litre of fuel consumed by a car (Figure 4).

Applying the strategies in the case of the hours-set (Figure 6c for Flexible Start and Figure 8c for Pause & Resume), we can see that the reductions reach 5.48% for Flexible Start and 10.14% for Pause & Resume. The reductions tend to increase with the increase in the value of the hours-set, while the reductions tend to be stable when varying the checking time. For HF-SCA, we also notice a nice plateau for Pause & Resume, while the Flexible Start strategy is less effective for such a long job.

In the case of the percentage set, the trend is similar to that of the hours set with a slightly lower gain.

5.2.4. Autoencoder

It has a similar duration to the IF workload, as well as similar power consumption and emissions. Consequently, the results of the applied strategies present a very similar pattern.

5.3. Average across All Regions

The previous benchmarks were calculated by varying the checking time and values for the time window, averaging the emission reduction throughout the entire year of 2021. We mentioned that the Flexible Start and Pause & Resume strategies were applied to the reference region. To provide comprehensive and reliable results, Table 3 shows the overall results achieved by the two strategies from [9], averaged across the seven European regions. It should be noted that in this case, only the hour-set was adopted because, from the previous subsections, it was evident that it is the more meaningful set, as in the case of the percentage set, workload durations of over 24 h would be required to obtain appreciable results.

Table 3. Emissions percentage reduction across all regions.

Algorithm	Strategy	Hours-Set			
		6	12	18	24
IF	Flexible Start Pause & Resume	3.91% 3.23%	4.50% 5.09%	5.20% 5.71%	6.05% 7.01%
SVM	Flexible Start Pause & Resume	3.07% 3.81%	5.17% 5.44%	6.29% 5.59%	7.51% 6.74%
HF-SCA	Flexible Start Pause & Resume	2.27% 1.48%	2.49% 3.20%	2.63% 4.88%	2.80% 4.99%
AE	Flexible Start Pause & Resume	3.97% 3.20%	4.90% 5.31%	5.39% 6.04%	6.56% 7.31%

Average carbon reductions for the two strategies are summarised in Table 4, where we can see that, in the scenario we considered, Pause & Resume performs slightly better than Flexible Start.

Table 4. Quantitative comparison between strategies.

Strategy	Avg Time Dilatation	Std Time Dilatation	Avg Carbon Reduction
Flexible Start	7:54 h	2:46 h	5.72%
Pause & Resume	11:15 h	3:26 h	6.51%

Finally, Table 4 also displays the average elongation in the duration across all workloads when subjected to the state-of-the-art strategies. Such values will be discussed in the next section.

6. Discussion

The aim of our research is to assess whether the results reported by Dodge et al. [9] in their paper can be confirmed for different regions and different kinds of workloads. For workloads, we considered four Anomaly Detection algorithms instanced in the financial domain (Fraud Detection). The four algorithms were trained using the grid search hyperparameters tuning strategy in order to have the worst case in energy consumption and training time.

When discussing the extent to which our findings are comparable with those of the state-of-the-art, we must consider that, unlike Dodge et al.[9]—whose considered workloads had length variability ranging from 0.3 to 216 h—our considered algorithms last in a much shorter range, i.e., from 2.5 to 16.0 h. Moreover, the marginal operating emissions for the considered region are very different too: 7–28 kg CO_2eq for Dodge et al. [9] (considering the BERT pretrain as a reference, lasting 36 h), vs. 0.6–2.7 kg CO_2eq in our study (considering the HF-SCA training, lasting 16 h). Because of these differences in the experiments, the workloads in Dodge et al. [9] had more room for improvement using the two carbon-aware strategies, hence reporting better results. In fact, having a wider MOER range—also within the same region—means that Flexible Start and Pause & Resume can choose far more favourable (re)starting times. Longer training times translate into higher carbon costs, but using reduction percentages rather than absolute values mitigates the risk of unfair comparisons.

6.1. Emission Reduction by Region

6.1.1. Flexible Start

When evaluating the Flexible Start algorithm for a fixed duration between 6 and 24 h, Dodge et al. [9] found significant emissions reductions for shorter jobs (e.g., the DenseNet experiments), with minimal savings for jobs longer than one day. In our experiment, we can confirm this observation: by looking at Table 3, we see that our longest workload (i.e., HF-SCA, lasting 16 h) achieves a maximum average of 2.80%, while shorter workloads reach 6–7%. At first glance, the size of this decrease seems much smaller: Dodge et al. [9] report on algorithms achieving around 26% of carbon savings (particularly the Dense experiments). However, if we consider experiments with similar duration (19 h for ViT tiny/small), carbon savings return a little more comparable, falling to approximately 5.5%. Nevertheless, percentages from the state-of-the-art double what we have observed in our experiments. This may be due to the different variability range of MOER, as mentioned at the beginning of this section. The regions we consider in this work are generally cleaner; hence, the effectiveness of any carbon-aware strategy is expected to be lower. However, it can be stated that Flexible Start provides robust carbon savings to longer workloads, even for cleaner regions.

For shorter workloads, we do not have directly comparable times: in our experiments, SVM, AE, and IF (in order of workload length) last around 2–4 h, while Dodge et al. [9] report on Bert finetune (6 h) and three DenseNets (20–25 min). Energy consumption is also not comparable (see Tables 1 and 3). Flexible Start provides a 14.5% carbon savings to the BERT finetune while achieving 26% for DenseNets as already outlined. In our experiments, we observe the same trend: the shorter the workload, the higher the savings provided by Flexible Start. Specifically, we have: IF, lasting 4:15 and saving 4.89%; AE, lasting 3:30 and saving 5.71%; SVM, lasting 2:30 and saving 6.67%. By the way, results are much less comparable in the case of short workloads. We can deduce that the effectiveness of Flexible Start highly depends on the greenness of the region.

6.1.2. Pause & Resume

When evaluating the Pause & Resume strategies for durations up to 100% of the duration of the original experiment, Dodge et al. [9] find the opposite of the Flexible Start result: short experiments like DenseNet 201 only see emissions reductions around 2–3%, while the 6.1B parameter transformer training run actually sees the largest decrease in emissions (11.4%). This observation remains valid in our experiment: Pause & Resume on HF-SCA achieves a 9.47% carbon saving, while shorter workloads are around 1–2%. Particularly, results are comparable between HF-SCA and the 6.1B transformers even if the workload sizes are vastly different (with the latter being more than 10 times longer than the former). What we have found is that the greener the region, the more effective Pause & Resume is (the opposite of Flexible Start).

6.2. Comparable Duration Increases

In a context where time is a valuable resource for the industry, paradoxically, the intention to reduce carbon emissions with strategies that extend the training time can contradict the need to obtain a trained model as quickly as possible. Indeed, Table 4 displays the average elongation in duration across all workloads when subjected to the state-of-the-art strategies. The table reports that the average time dilation for each workload resulting from the application of the state-of-the-art strategies is around 11 h for Pause & Resume and nearly 8 h for Flexible Start. This confirms what Dodge et al. [9] reported: we can think of the Flexible Start algorithm as a version of Pause & Resume where there is only one start time and no pausing is allowed. Thus, Flexible Start results in a lower bound than Pause & Resume, as shown in Table 4: Pause & Resume can only perform better than Flexible Start.

However, such strategies are not designed to achieve results within a time window equal to the length of the training. In addition to the benefits of reduced carbon emissions, it is important to consider the time saved, which translates into profitability for the industry. Other strategies not subjected to time dilation should be investigated, like those shifting the workload spatially instead of temporally. Such approaches could constrain the training start time to the one determined by the decision-maker while reducing emissions through workload relocation.

6.3. Final Considerations

The two benchmark strategies can reduce the impact of AI training through simple modifications to the training scripts. The effectiveness of these interventions may vary depending on the region, the type, and the duration of the workload, but the overall impact is consistently positive. This raises the question of why these strategies have not been widely adopted by the AI community, which includes practitioners, industry, and academia. We concur with Dodge et al. [9], who urge researchers and practitioners to record and report the emissions associated with AI projects. However, they also recognise that some projects operate under strict time constraints, and implementing these strategies could significantly delay progress, potentially leading to increased emissions in other areas of the project. Table 4 shows an average delay of approximately 8 h for Flexible Start and approximately 11 h for Pause & Resume. This is a considerable sacrifice, effectively quadrupling the training time for the industry. Nevertheless, this issue primarily affects cases where frequent re-training is necessary to keep up with changes in data, a phenomenon known as concept drift [49]. Further research is needed to evaluate how sustainable concept drift adaptation techniques—such as those based on sliding windows or specific data change monitoring tools [50–52]—can outperform full-history approaches in mitigating the side effects of time-based carbon-aware training strategies.

6.4. Comparison with Existing Literature

Basing on the state-of-the-art reported in Section 2, the aim of this subsection is to show how this work contributes to advancing the current knowledge in the field of observational Green AI. Particularly, we position our work within the relevant body of literature in order to compare it with existing studies, updating the pre-existing mapping with our original contribution.

As it is evident from Table 5, our work closely aligns with the characteristics of the previous work from Dodge et al. [9] and also adds a further point of view on the topic of observational Green AI. Particularly, our work uses a tabular dataset to train algorithms outside of the NLP and computer vision domains. Moreover, we contribute to the field by providing (upon requests) the script to replicate our results.

Table 5. Comparison with the state-of-the-art.

Paper	Year	Green AI Definition	Venue Type	Study Type	Topic	Domain	Type of Data	Artifact Considered	Considered Phase	Validation Type	Considered System Size	Saving	Industry Involvement	Intended Reader	Providing Tool
This work	2024	carbon footprint	journal	observational	deployment, monitoring	cloud	text	all	training	field experiment	635K	5.72 — 6.51%	Mix	Academic	Yes
Strubell et al. [7]	2019	ecological footprint, carbon footprint, energy efficiency	conference	observational	monitoring	general	NLP	algorithm - deep neural network	training	-	-	Academic	Academic	No	
Dodge et al. [9]	2022	carbon footprint	conference	observational	deployment, monitoring	cloud	NLP, image	all	all	field experiment	up to 6B	3.7 — 8.7%	Mix	Academic	No
Fraga-Lamas et al. [16]	2021	energy efficiency, carbon footprint	journal	observational	deployment, emissions, network-architecture	edge	image	all	all	laboratory	-	-	Academic	Academic	No
Wu et al. [17]	2022	ecological footprint, carbon footprint	conference	observational	monitoring	general	-	algorithm - general	all	field experiment	-	-	Academic	Academic	No
Ferro et al. [18]	2021	carbon footprint, energy efficiency	journal	observational	monitoring, model-comparison	general	number, text	algorithm - decision tree	all	laboratory	5M	-	Academic	Academic	No
Asperti et al. [19]	2021	energy efficiency, carbon footprint	journal	observational	model-comparison	general	image	algorithm - deep neural network	training	laboratory	10K	-	Academic	Academic	No
Jooste et al. [20]	2022	carbon footprint	journal	observational	hyperparameter-tuning	general	text	algorithm - deep neural network	training	laboratory	2M	50%	Academic	Academic	No
Bannour et al. [21]	2021	carbon footprint	workshop	observational	model-comparison, monitoring	general	NLP	algorithm	all	laboratory	1.2K	-	Academic	Mix	No

6.5. Threat to Validity

In this work, we have made some assumptions that may limit the validity of the presented results. The primary threat to internal validity is that we compared training workloads on a machine different from those used in the considered state-of-the-art study. To mitigate this, we evaluated the workloads based on absolute energy consumption rather than solely on execution times. The main threat to external validity arises from considering a limited set of AI workloads, each lasting less than one day. To mitigate this threat, in the discussion section we have considered this difference when performing a careful comparison with the existing experiments retrieved from the state-of-the-art, which instead lasts more than 24 h. The main threat to construct validity consists in not having considered the average number of pauses per hour as a metric to fulfil a complete comparison between the Pause & Resume experiments. To mitigate this, we included a dedicated subsection in the discussion section to explore the implications of the time elongation associated with this strategy.

7. Conclusions

In this paper, we have replicated experiments from a state-of-the-art study on carbon-aware AI training strategies [9]. These strategies, known as Flexible Start and Pause & Resume, leverage time resources to reduce the carbon footprint of training workloads. Their names are self-explanatory, as they schedule energy-intensive computations during optimal times, based on the Marginal Operating Emissions Rate (MOER). While the previous study demonstrated the effectiveness of these strategies in specific scenarios and highlighted their straightforward implementation for industry adoption, further research was necessary to evaluate their effectiveness across different scenarios. This research aims to determine the extent to which emission reductions are consistent in a different context.

Particularly, while previous research focused on NLP and Computer Vision, our study uses the Fintech domain as a benchmark, defining a testbed of four algorithms for the fraud detection task: Isolation Forest, Support Vector Machine, autoencoder, and HF-SCA (a highly customised U-Net). The workloads in our study differ in their shorter duration and the geographical regions considered, specifically aligning with the presence of AWS data centres in Europe. It is crucial to note that the regions we studied are generally greener than those considered in previous research.

Our results are generally consistent with existing literature, though we observed some notable differences in the effectiveness of the strategies. For Flexible Start, we confirm that shorter workloads yield higher carbon savings. However, the percentages observed in our experiments are about half of those reported in state-of-the-art studies, which may be due to the different variability ranges of MOER in the regions considered. For Pause & Resume, the greener the region, the more effective the strategy becomes. In fact, the results for HF-SCA are comparable to those for a 6.1B transformer, despite the latter's workload being more than ten times longer.

We can summarise the main outcomes of our research as follows:

- We reimplemented the code for Flexible Start and Pause & Resume, which is available upon request.
- We confirmed the effectiveness of two existing carbon-aware AI training strategies, providing additional useful insights.
- We contributed to the field of Green AI, hopefully encouraging further research efforts in this specific domain.

The results of our research are intended to benefit the following categories of stakeholders:

- Industry: our findings provide more evidence on the effectiveness of the proposed strategies, enabling industry players to decide if, when, and how to apply these strategies to make their AI usage greener and more aligned with the ecological transition, particularly with regard to ESG requirements.

- AI Community: the results can be used to develop AI libraries and frameworks that natively support carbon-aware scheduling of training workloads.
- Researchers in Green AI and Green Software: researchers can use our results to assess and compare the effectiveness of the proposed strategies in new scenarios and to develop and benchmark new strategies against those analysed in this work.

Further research is needed to enhance our understanding of carbon-aware training strategies. Specifically, similar analyses should be applied to strategies involving the spatial relocation of workloads, as relying solely on a specific region limits the potential carbon savings. Another interesting aspect is the potential of these strategies to mitigate the carbon emissions of the inference phase; in this case, the same analysis would reveal the extent to which the strategies could make AI models deployed in production greener. Additionally, the implications of delaying user responses due to unfavourable MOER at specific times should be considered, particularly concerning the time sensitivity of different domains. Finally, the usefulness of concept drift adaptation techniques on mitigating the need to retrain AI models should be measured in order to make temporally shifting workloads more economically sustainable for the industry.

Author Contributions: Conceptualization, L.M. and R.V.; methodology, R.V.; formal analysis, R.V.; writing—original draft preparation, R.V.; writing—review and editing, L.M.; visualization, R.V.; supervision, L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Scripts and data are available upon direct request to the authors.

Acknowledgments: This work is one of the results of the AMEDEA (Assessment and Mitigation of the Environmental Impact of Deep Learning Algorithms) project, which was supported by CINECA within the context of the ISCRA Class C programme (cod. IsCa7_AMEDEA). The authors want to thank the research partner WattTime for providing the historical dataset of local-marginal emissions used for this work. We also thank Alessio Errico for providing the data for this confirmation study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
AD	Anomaly Detection
AWS	Amazon Web Services
NLP	Natural Language Processing
MOER	Marginal Operating Emission Rate
IF	Isolation Forest
SVM	Support Vectors Machine
HF-SCA	Hands-Free Strong Customer Authentication
AE	Autoencoder
AUC	Area Under the ROC Curve
GDPR	General Data Protection Regulation
ESG	Environment Social Governance

References

1. Aslan, J.; Mayers, K.; Koomey, J.G.; France, C. Electricity intensity of internet data transmission: Untangling the estimates. *J. Ind. Ecol.* **2018**, *22*, 785–798. [CrossRef]
2. Mainetti, L.; Aprile, M.; Mele, E.; Vergallo, R. A sustainable approach to delivering programmable peer-to-peer offline payments. *Sensors* **2023**, *23*, 1336. [CrossRef] [PubMed]
3. Masanet, E.; Shehabi, A.; Lei, N.; Smith, S.; Koomey, J. Recalibrating global data center energy-use estimates. *Science* **2020**, *367*, 984–986. [CrossRef] [PubMed]

4. Vergallo, R.; D'Alò, T.; Mainetti, L.; Paiano, R.; Matino, S. Evaluating Sustainable Digitalization: A Carbon-Aware Framework for Enhancing Eco-Friendly Business Process Reengineering. *Sustainability* **2024**, *16*, 7789. [CrossRef]
5. Corcoran, P.; Andrae, A. *Emerging Trends in Electricity Consumption for Consumer ICT*; National University of Ireland: Galway, Ireland, 2013.
6. Tamburrini, G. The AI carbon footprint and responsibilities of AI scientists. *Philosophies* **2022**, *7*, 4. [CrossRef]
7. Strubell, E.; Ganesh, A.; McCallum, A. Energy and policy considerations for modern deep learning research. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13693–13696.
8. Taylor, J.A.; Callaway, D.S.; Poolla, K. Competitive energy storage in the presence of renewables. *IEEE Trans. Power Syst.* **2012**, *28*, 985–996. [CrossRef]
9. Dodge, J.; Prewitt, T.; des Combes, R.T.; Odmark, E.; Schwartz, R.; Strubell, E.; Buchanan, W. Measuring the carbon intensity of AI in cloud instances. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 1877–1894.
10. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *Acm Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [CrossRef]
11. Verdecchia, R.; Sallou, J.; Cruz, L. A systematic review of Green AI. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2023**, *13*, e1507. [CrossRef]
12. Verdecchia, R.; Lago, P.; Ebert, C.; De Vries, C. Green IT and green software. *IEEE Softw.* **2021**, *38*, 7–15. [CrossRef]
13. Wiedmann, T.; Minx, J. A definition of "carbon footprint". *Ecol. Econ. Res. Trends* **2008**, *1*, 1–11.
14. Matuštík, J.; Kočí, V. What is a footprint? A conceptual analysis of environmental footprint indicators. *J. Clean. Prod.* **2021**, *285*, 124833. [CrossRef]
15. Luccioni, A.S.; Hernandez-Garcia, A. Counting carbon: A survey of factors influencing the emissions of machine learning. *arXiv* **2023**, arXiv:2302.08476.
16. Fraga-Lamas, P.; Lopes, S.I.; Fernández-Caramés, T.M. Green IoT and edge AI as key technological enablers for a sustainable digital transition towards a smart circular economy: An industry 5.0 use case. *Sensors* **2021**, *21*, 5745. [CrossRef] [PubMed]
17. Wu, C.J.; Raghavendra, R.; Gupta, U.; Acun, B.; Ardalani, N.; Maeng, K.; Hazelwood, K. Sustainable ai: Environmental implications, challenges and opportunities. *Proc. Mach. Learn. Syst.* **2022**, *4*, 795–813.
18. Ferro, M.; Silva, G.D.; de Paula, F.B.; Vieira, V.; Schulze, B. Towards a sustainable artificial intelligence: A case study of energy efficiency in decision tree algorithms. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e6815. [CrossRef]
19. Asperti, A.; Evangelista, D.; Loli, Piccolomini, E. A survey on variational autoencoders from a green AI perspective. *SN Comput. Sci.* **2021**, *2*, 301. [CrossRef]
20. Jooste, W.; Haque, R.; Way, A. Knowledge distillation: A method for making neural machine translation more efficient. *Information* **2022**, *13*, 88. [CrossRef]
21. Bannour, N.; Ghannay, S.; Névéol, A.; Ligozat, A.L. Evaluating the carbon footprint of NLP methods: A survey and analysis of existing tools. In Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, Onilne, 10 November 2021; pp. 11–21.
22. Kaack, L.H.; Donti, P.L.; Strubell, E.; Kamiya, G.; Creutzig, F.; Rolnick, D. Aligning artificial intelligence with climate change mitigation. *Nat. Clim. Chang.* **2022**, *12*, 518–527. [CrossRef]
23. Willenbacher, M.; Hornauer, T.; Wohlgemuth, V. Rebound effects in methods of artificial intelligence. In *Environmental Informatics*; Springer International Publishing: Cham, Switzerland, 2021; pp. 73–85.
24. Patterson, D.; Gonzalez, J.; Hölzle, U.; Le, Q.; Liang, C.; Munguia, L.M.; Dean, J. The carbon footprint of machine learning training will plateau, then shrink. *Computer* **2022**, *55*, 18–28. [CrossRef]
25. Ligozat, A.L.; Lefevre, J.; Bugeau, A.; Combaz, J. Unraveling the hidden environmental impacts of AI solutions for environment life cycle assessment of AI solutions. *Sustainability* **2022**, *14*, 5172. [CrossRef]
26. Dhar, P. The carbon impact of artificial intelligence. *Nat. Mach. Intell.* **2020**, *2*, 423–425. [CrossRef]
27. Rohde, F.; Gossen, M.; Wagner, J.; Santarius, T. Sustainability challenges of artificial intelligence and policy implications. *öKologisches -Wirtsch.-Fachz.* **2021**, *36*, 36–40. [CrossRef]
28. Wenninger, S.; Kaymakci, C.; Wiethe, C.; Römmelt, J.; Baur, L.; Häckel, B.; Sauer, A. How sustainable is machine learning in energy applications?–The sustainable machine learning balance sheet. In Proceedings of the International Conference on Wirtschaftsinformatik (WI) 2022, Online, 21–23 February 2022.
29. Henderson, P.; Hu, J.; Romoff, J.; Brunskill, E.; Jurafsky, D.; Pineau, J. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–43.
30. Anthony, L.F.W.; K.; ing, B.; Selvan, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv* **2020**, arXiv:2007.03051.
31. Lannelongue, L.; Grealey, J.; Inouye, M. Green algorithms: Quantifying the carbon footprint of computation. *Adv. Sci.* **2021**, *8*, 2100707. [CrossRef]
32. Jurj, S.L.; Opritoiu, F.; Vladutiu, M. Environmentally-friendly metrics for evaluating the performance of deep learning models and systems. In *Neural Information, Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, 23–27 November 2020*; Proceedings, Part III 27; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 232–244.
33. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.

34. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
35. Distante, C.; Fineo, L.; Mainetti, L.; Manco, L.; Taccardi, B.; Vergallo, R. HF-SCA: Hands-Free Strong Customer Authentication Based on a Memory-Guided Attention Mechanisms. *J. Risk Financ. Manag.* **2022**, *15*, 342. [CrossRef]
36. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
37. Sandbox Regolamentare, Banca d'Italia. Retrieved June 2024. Available online: https://www.bancaditalia.it/focus/sandbox (accessed on 11 September 2024).
38. NumPy, Jim Hugunin. Retrieved June 2024. Available online: https://numpy.org/doc/stable/ (accessed on 11 September 2024).
39. Pandas, Wes McKinney, J. Brock Mendel, Joris Van den Bossche e Jeff Reback. Retrieved June 2024. Available online: https://pandas.pydata.org (accessed on 11 September 2024).
40. Cournapeau, D.; Grisel, O.; Varoquaux, G.; Mueller, A.G.E.A. Scikit-Learn Machine Learning in Python. Retrieved June 2024. Available online: https://scikit-learn.org (accessed on 11 September 2024).
41. Tensorflow, Google Brain. Retrieved June 2024. Available online: https://www.tensorflow.org (accessed on 11 September 2024).
42. PyTorch, Meta AI. Retrieved June 2024. Available online: https://pytorch.org (accessed on 11 September 2024).
43. Track and Reduce CO_2 Emissions from Your Computing, Clever Cloud. Retrieved June 2024. Available online: https://codecarbon.io/ (accessed on 11 September 2024).
44. Hunter, J.D. Matplotlib: Visualization with Python. Retrieved June 2024. Available online: https://matplotlib.org (accessed on 11 September 2024).
45. Jagannadharao, A.; Beckage, N.; Nafus, D.; Chamberlin, S. Timeshifting strategies for carbon-efficient long-running large language model training. *Innov. Syst. Softw. Eng.* **2023**, 1–15. [CrossRef]
46. Wiesner, P.; Behnke, I.; Kilian, P.; Steinke, M.; Kao, O. *Vessim: A Testbed for Carbon-Aware Applications and Systems*; Technische Universität Berlin: Berlin, Germany, 2024.
47. Ross Fairbanks. Retrieved June 2024. Available online: https://rossfairbanks.com/2023/07/12/carbon-aware-spatial-shifting-with-karmada/ (accessed on 11 September 2024).
48. Retrieved June 2024. Available online: https://www.wonderingchimp.com/exploring-the-green-apis/ (accessed on 11 September 2024).
49. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–37. [CrossRef]
50. Poenaru-Olaru, L.; Karpova, N.; Cruz, L.; Rellermeyer, J.; van Deursen, A. Maintenance Techniques for Anomaly Detection AIOps Solutions. *arXiv* **2023**, arXiv:2311.10421.
51. Järvenpää, H.; Lago, P.; Bogner, J.; Lewis, G.; Muccini, H.; Ozkaya, I. A synthesis of green architectural tactics for ml-enabled systems. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Societ, Lisbon, Portugal, 14–20 April 2024; pp. 130–141.
52. Poenaru-Olaru, L.; Sallou, J.; Cruz, L.; Rellermeyer, J.S.; Van Deursen, A. Retrain AI Systems Responsibly! Use Sustainable Concept Drift Adaptation Techniques. In Proceedings of the 7th International Workshop on Green And Sustainable Software (GREENS), Melbourne, Australia, 14 May 2023; pp. 17–18.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.