




A Data-Centric Approach for Reducing Carbon Emissions in Deep Learning

Martín Anselmo and Monica Vitali^(✉) 

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
Milan, Italy

martinfelix.anselmo@mail.polimi.it, monica.vitali@polimi.it

Abstract. The growing popularity of Deep Learning (DL) in recent years has had a large environmental impact. Training models require a lot of processing and computation and therefore require a lot of energy. The size of these models and the amount of data required for training them have grown exponentially, not comparable to the performance improvements. Recently, some model-centric approaches have been proposed to limit the environmental impact of AI. This paper complements them by proposing a data-centric “Green AI” approach, focusing on the data preparation phase of the DL pipeline. A general methodology, valid for any DL task, is proposed. This methodology is based on analyzing data characteristics, mainly the data quality and volume dimensions, and observing how these affect carbon emissions and performance on different models. With this information, a human-in-the-loop (HITL) approach is provided to support researchers in obtaining a modified and reduced version of a dataset that can decrease the environmental impact of training while achieving a specified performance goal. To demonstrate its validity, the proposed methodology is applied to the time series classification task and a prototype has been developed which demonstrates the possibility of reducing the carbon emissions of DL training by up to 50%.

Keywords: Data-centric AI · Green AI · Data Preparation · Data Quality · Deep Learning · Big Data

1 Introduction

In recent years, Deep Learning (DL) has become very popular for extracting knowledge from non-structured data, such as images or time series. The increase in computing power made possible with the development of new computing hardware and new deep neural network architectures has allowed for unprecedented results in previously complex tasks. From 2012 to 2019, the computing power required by state-of-the-art results has increased by $300,000 \times$ [17]. More recently, there has been an even bigger increase with models such as GPT-3, which has 175 billion parameters and was trained on a dataset of nearly a trillion words.

This rise of DL results in a huge environmental impact since the hardware required is very power-hungry. Recently, a distinction between “Red AI” and

“Green AI” has been introduced in [17]: the former refers to Artificial Intelligence (AI) research focusing on performance aspects only, the latter refers to environmentally-aware research on AI. Most of the research on “Green AI” has been addressed to developing better and more efficient algorithms and architectures. However, DL training is always preceded by a data preparation phase, in charge of preparing the dataset for the training task. Information Systems Engineering (ISE) expertise can be beneficial in the design of the data preparation phase and can impact the overall energy consumption of the DL task.

This work aims at complementing the existing model-centric approaches with a data-centric approach. We propose a methodology that improves data preparation by transforming the original training set such that: (i) the performance constraints of the resulting model are satisfied; (ii) the environmental impact of the training phase is reduced. These goals are reached by taking into account the characteristics of the dataset, such as data volume and Data Quality (DQ).

The proposed methodology is validated on time series classification using Deep Neural Networks (DNNs). The prototype of a tool that researchers interested in reducing their carbon emissions in this task can use is also provided.

The paper is organized as follows. Section 2 describes existing work. Section 3 motivates the approach and set the goals of the methodology. Section 4 and Sect. 5 introduce an architecture for Data-centric Green AI and describe an implementation in the context of time series classification. Section 6 validates the methodology, while Sect. 7 summarizes the approach and outlines future developments.

2 State of the Art

AI has become pervasive in all fields, and its strong interdependency with climate change has been demonstrated [16]. Several applications of AI can play a role in the reduction of the effects of climate change. At the same time, AI is also the application affecting the environmental impact of IT the most. In 10 years, the computing power required by AI has increased 300'000 times [10]. DL is a subset of AI which uses DNNs as predictive models. The learning process requires several iterations and can take many hours or days, on very power-hungry hardware. Models such as Convolutional Neural Networks (CNNs), Fully Convolutional Networks (FCNs), and Residual Networks (ResNet) have proved to be very effective in terms of performance at the expense of a relevant environmental impact [7, 8]. The most power-hungry phase in DL is the hyperparameter (HP) search, since it considers the training of many models with different configurations to find the one with the best performance [20].

The issue of the environmental impact of AI has been discussed in [17], introducing and comparing the two opposite concepts of “Red AI” and “Green AI”. The former refers to a performance-focused approach, where all the efforts are put into accuracy, disregarding costs and efficiency. The second envisions a more sustainable approach to AI, encouraging a reduction in resources spent. The main aspects to consider for reducing the environmental impact of AI are analyzed, focusing mainly on architectural and algorithm-related aspects.

This can also be seen in [25] and [15], where the environmental impact of DL is considered focusing on the infrastructural, architectural, and location aspects. They partially consider the data perspective through transfer learning and active learning approaches. In [6], authors focus instead on the environmental impact of the model selection and the hyperparameter search. As shown in [4], modelling the DL task is only one step, preceded by a data preparation phase, which might affect as well the environmental impact of the overall task.

Data preparation is essential in many contexts for the analysis of large volumes of data [13, 14]. Data preparation is the preliminary phase of every DL task, which can improve the resulting model performance [11, 19] or affect the dataset balance [22]. Data preparation can also affect the environmental impact of DL tasks. The main factor to consider is data volume [12], affecting the training time and the number of resources needed, with sometimes marginal effects in terms of performance [21]. A preliminary data-centric empirical study on Green AI [23] has shown that modifications on the volume of datasets can drastically reduce energy consumption, with a limited decline in accuracy. Data selection should be DQ-driven. The data preparation step in the AI lifecycle is necessary to prevent incorrect results and biases due to poor quality data [2]. A study on the effect of DQ issues on several ML models have been performed in [3], where completeness, accuracy, consistency, completeness, and class balance have been considered, suggesting a limited relevance of class balancing on the model performance as far as the balance is higher than or equal to the original dataset.

This paper takes a data-centric perspective to Green AI to complement existing model-centric approaches with improved training data management.

3 Motivation and Goals for Data-Centric Green AI

AI is a first-class citizen in modern data centers, and the amount of computational and storage resources employed for supporting AI has been increasing and keeps growing. AI applications have become a utility, as demonstrated by the wide and continuously increasing adoption in different fields and for diverse purposes. Current approaches to AI focus on performance optimization and consider sustainability mainly from a model-centric perspective. The availability of huge datasets has enabled the training of complex models and boosted their performance. However, the data size used for training significantly affects the time and the resources needed for the training, impacting the environmental sustainability of AI applications [20]. Not all data have the same relevance for building the model: good quality datasets are necessary for creating high-quality models able to perform accurate predictions [9]. This problem is amplified by big data [5]. This paper adopts a sustainability-driven perspective on DL, with a data-centric focus: the environmental impact of DL applications is reduced by selecting a proper subset of the data for training the model while ensuring a required performance level. This paper identifies three incremental goals:

- **Goal 1:** Explore which data-centric characteristics of DL pipelines contribute the most to energy usage, and find out which can be tweaked so the overall environmental impact is reduced.

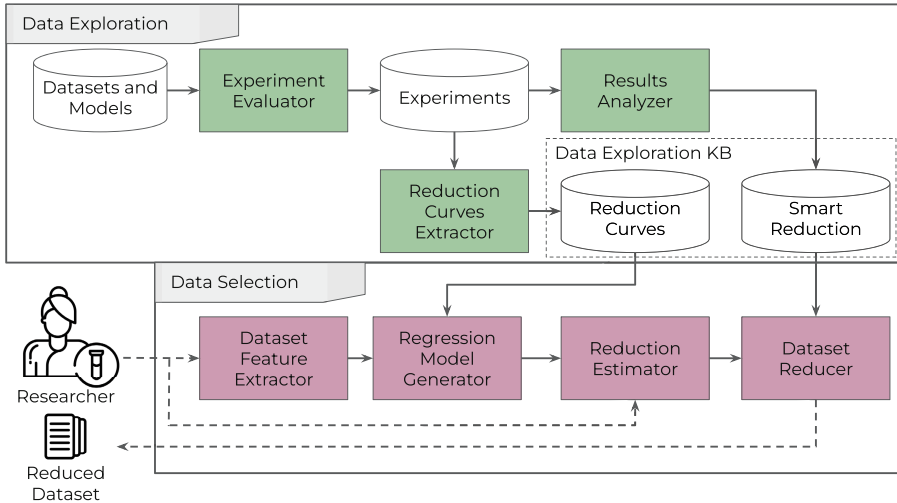


Fig. 1. Architecture for Data-Centric Green DL

- **Goal 2:** Model the relation between the discovered data-centric characteristics and the resulting model’s performance.
- **Goal 3:** Reduce the DL impact on carbon emissions by making more efficient use of data while being constrained by performance requirements.

To reach them, a general and data-centric methodology valid for any DL task is proposed, including two phases: a **Data Exploration Phase** in charge of reaching the first two goals through the generation of a knowledge base, and a **Data Selection Phase** focused on the third goal. The proposed approach can be integrated with existing model-centred techniques to improve AI sustainability.

4 An Architecture for Data-Centric Green DL

In this section, we present a detailed architecture for supporting Data-Centric Green DL in Fig. 1. The architecture is split into two parts, corresponding to each one of the phases. Since each DL task has different characteristics (i.e., models, algorithms, datasets and performance metrics), the actual implementation of the architecture depends on the specific DL task to support. To validate our approach, an implementation for time series classification is presented in Sect. 5.

4.1 Data Exploration

The **Data Exploration** part of the architecture focuses on the components required for addressing Goal 1 and Goal 2. The output is a *Data Exploration KB* containing information about how an experiment (defined here as a dataset-model pair) is affected by the manipulation of a data-centric characteristic.

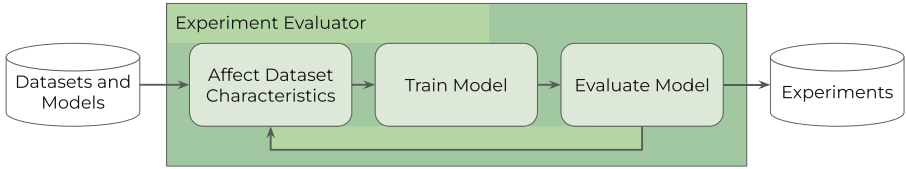


Fig. 2. Overview of the Experiment Evaluator component behaviour

The performance of the resulting model and the carbon emissions generated during training will be considered. This part consists of three components.

The *Experiments Evaluator* executes a set of experiments to collect useful data and learn the trade-offs between data volume vs performance, data volume vs emissions, and DQ vs performance. More specifically, the *Experiments Evaluator* runs a set of experiments, defined as:

$$exp = \langle mod, ds \rangle \quad (1)$$

where *mod* is a specific DL model and *ds* is a dataset for training the model. For each experiment, the *Experiments Evaluator* runs a set of sub-experiments, changing its configurations. A sub-experiment is defined as:

$$sub_exp = \langle exp, [conf] \rangle \quad (2)$$

where *[conf]* is the set of configurations to test (data volume or DQ), each one identifying a specific aspect and a specific value for that aspect (e.g., *volume = 50%*). In order to isolate side effects, only one configuration is tested at each time and for each aspect, several values are tested. For each sub-experiment, the resulting modified dataset is used to train the model and a set of performance metrics is evaluated and stored. The overall process is shown in Fig. 2. The component uses as input a set of models and relative datasets stored in the *Datasets and Models* DB. The output of the component is stored in the *Experiments* DB as a table containing the following information for each sub-experiment:

$$exp_res = \langle ds, mod, [conf], [perf] \rangle \quad (3)$$

where *[perf]* is the set of performance metrics evaluated with their assessed values. The set of metrics to evaluate depends on the task (e.g., recall, precision, accuracy, F1-score for classification tasks).

The data stored in the *Experiments* DB have a dual use. The *Results Analyzer* component analyzes the impact of data volume and DQ on the model performance. It accesses the experiments with different configurations involving DQ aspects and provides a ranking of which DQ metric degradation mostly affects the model performance. It also validates the carbon emission reduction capabilities for each configuration to detect which data aspects mostly affect CO₂ emissions. This information is stored in the *Smart Reduction* DB.

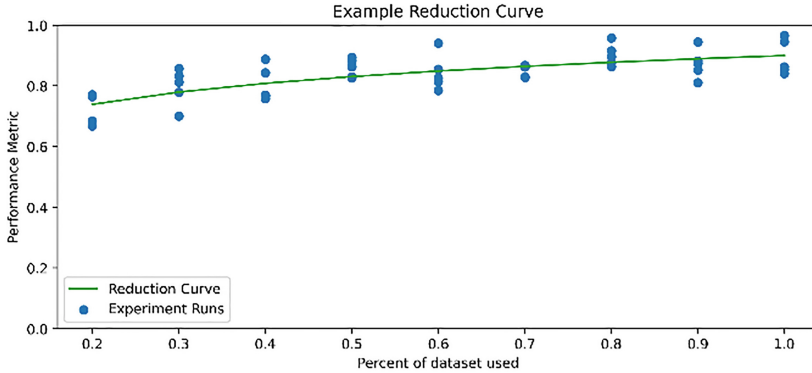


Fig. 3. Sample reduction curve for an experiment

The *Reduction Curves Extractor* component focuses on the data volume vs performance trade-off and aggregates the information collected by the *Experiment Evaluator* to build a reduction curve for each experiment. An example can be seen in Fig. 3, in which actual data are collected only for nine configurations of the data volume, while the generated curve enables us to estimate which will be the performance metric also for intermediate configurations.

All the activities in this phase are executed only once and aim at collecting information to enable the **Data Selection Phase**.

4.2 Data Selection

The **Data Selection** part of the architecture exploits the results collected in the previous phase to support a researcher willing to execute a new training task in reducing its environmental impact.

The most relevant component of this part of the architecture is the *Regression Model Generator*. At first, this component uses the data stored in the *Reduction Curves* DB to train a predictive regression model that will be able to build a new reduction curve for an unseen experiment starting from the reduction curves examples contained in the DB. Once the regression model is built, it can be used to perform a prediction every time a researcher submits a new experiment.

The *Regression Model Generator* is the only component of this part of the architecture running partially in batch mode. All other components run interactively, providing a human-in-the-loop (HITL) approach. The researcher is, in fact, in charge of providing some preliminary information to the system. The information provided by the researcher belongs to four different categories:

- **Dataset Information:** $DI = \langle ds, d.type \rangle$. The researcher provides the dataset ds for training the model and specifies the data type $d.type$ from a list of supported types (e.g., image, sensor data, etc.).
- **Model Information:** $MI = \langle arch.type, \#par \rangle$. The researcher provides the features of the model to be trained, consisting in the type of architecture

arch.type, selected from a list of available architectures, and in the number of parameters of the model *#par*;

- **Baseline Execution Information:** $\mathcal{BI} = \langle ds_p, perf_{val} \rangle$. The researcher provides the results of a preliminary execution of the experiment using a randomly reduced dataset. More specifically, the researcher provides the tested dataset size ds_p and the obtained performance value with that size $perf_{val}$;
- **Performance Goal:** $\mathcal{G} = \langle perf_{metric}, perf_{val} \rangle$ the researcher sets the minimum acceptable value $perf_{val}$ for a specific performance metric $perf_{metric}$.

The inputs provided by the researcher are used by the different components of the architecture. The dataset ds is first processed by the *Dataset Features Extractor* component, which performs profiling activities to extract metadata and compute DQ metrics about the dataset. The enriched dataset information \mathcal{DI}' and the model information \mathcal{MI} are used by the *Regressor Model Generator* that matches them with the parameters and configurations of its internal model and predicts a regression curve for the new experiment. With this curve and \mathcal{G} , the *Reduction Estimator* suggests the volume of data \hat{p} that ensures \mathcal{G} while reducing energy consumption. The *Dataset Reducer* extracts a subset $ds_{\hat{p}} \subset ds$ of size \hat{p} exploiting the information provided in the *Smart Reduction DB* about the DQ metric ranking. As an output, the researcher gets the *Reduced Dataset* $ds_{\hat{p}}$, with higher DQ and lower data volume, that can be used to perform a new training with a limited environmental impact.

5 Implementation of the Architecture

The actual implementation of the architecture presented in Sect. 4 depends on the specific DL task to be addressed. To demonstrate it, we describe its implementation for the time series classification task. In this context, we can define a *dataset* ds as a collection of *data points* DP , where each data point dp is a time series consisting of L values collected over a time period.

A collection of datasets and models have been used to implement the **Data Exploration Phase** and stored in the *Dataset and Models* DB:

- the datasets are selected from the UCR/UEA repository¹, consisting of over 100 datasets with different characteristics over a variety of fields;
- three different architectures - MLP, FCN, and ResNet [24] were used.

For the sake of simplicity, we limited our evaluation to a single performance metric, and we selected the F1-Score, which represents both the correctly classified series (precision) and the incorrect ones (recall).

The experiments were run on Google Colab², on Intel(R) Xeon(R) CPU and a Nvidia Tesla T4 GPU instances. Carbon emissions were measured in KgCO₂ with CodeCarbon³, manually setting the execution in Italy to reduce variability. All the code is freely available on GitHub⁴.

¹ <https://www.timeseriesclassification.com/dataset.php>.

² <https://colab.research.google.com>.

³ <https://codecarbon.io/>.

⁴ <https://github.com/mfanselmo/Time-Series-Classification-GreenAI>.

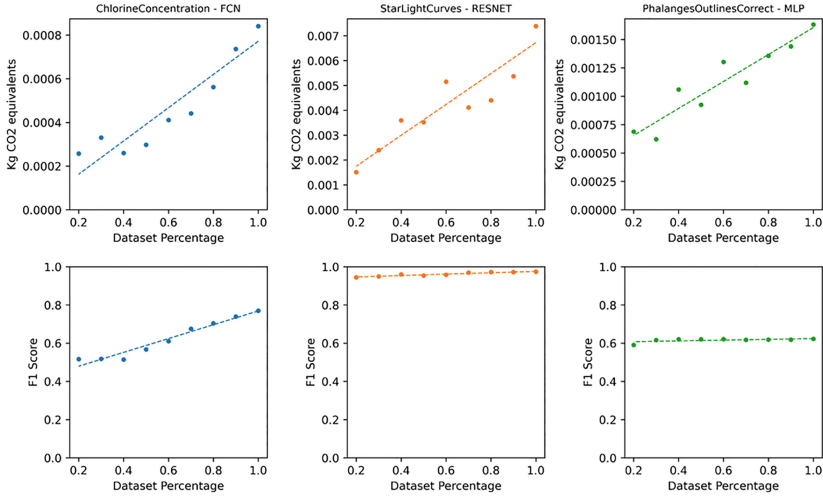


Fig. 4. Volume and carbon emissions (top) or F1-Score (bottom) trade-off

5.1 Data Exploration Implementation

As described in Sect. 4.1 and depicted in Fig. 2, several experiments are executed combining the datasets and models contained in the *Dataset and Models* DB and storing the results in the *Experiments* DB. In each sub-experiment a different dataset configuration was tested, considering two aspects:

- **Data Volume:** from 100% all the way down to 20%, in steps of 10%. At this stage, data points are selected randomly from the dataset;
- **DQ:** injecting errors on accuracy, consistency, and completeness, from 1 to 0.2 in steps of 0.1. To obtain the dirty dataset, we apply *data pollution* as described in [3]: for each DQ metric and for each step, the set of data points to pollute is randomly extracted, and the data points are properly modified:
 - Accuracy: it is computed as the percentage of data points with a correct target value associated. For each of the selected data points, the target value is substituted with a different one;
 - Completeness: it is computed as the complement of the percentage of missing values in the time series composing the dataset. For each of the selected data points, values of the time series are randomly removed;
 - Consistency: it is computed as the percentage of data points that follow the consistency rule: two series with the same values must be associated with the same target value. Each of the selected data points is duplicated and a different target value is assigned to the copy.

For each configuration, five experiments are executed to reduce noise for a total of 1'215 experiments.

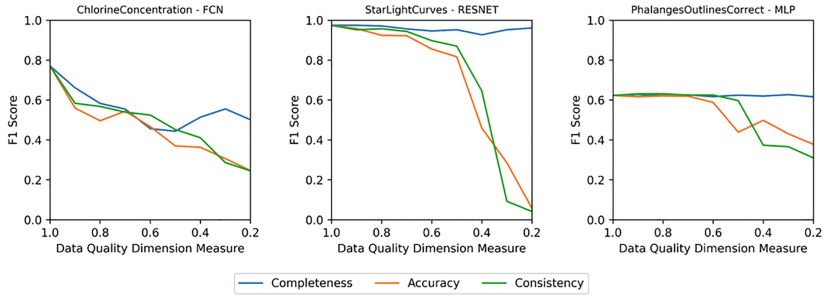


Fig. 5. Impact of different DQ dimensions on model performance

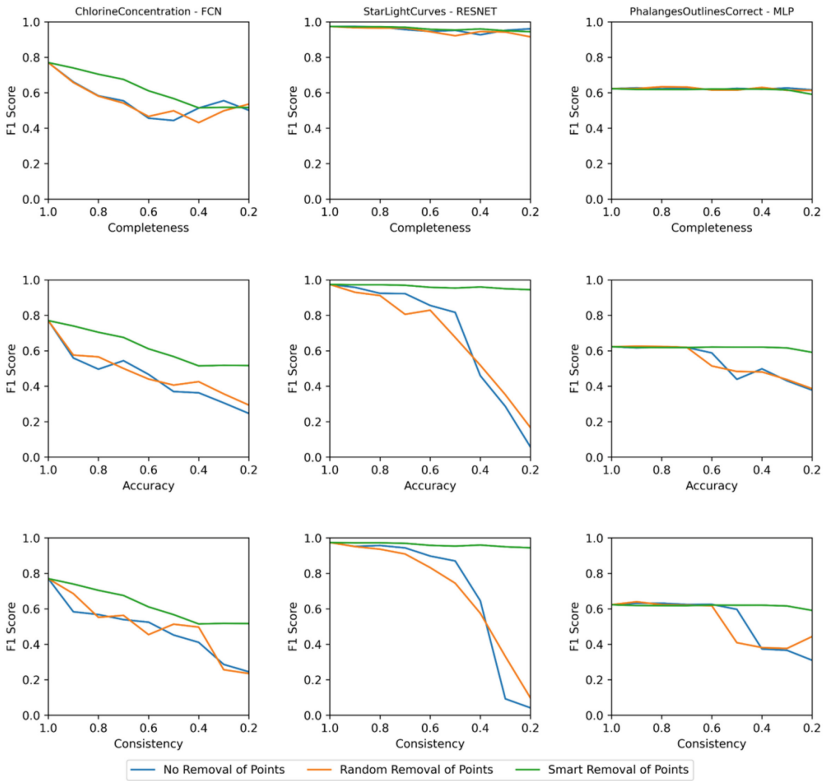


Fig. 6. Comparison of the impact on the performance of different data reduction strategies: smart removal, random removal, no removal

The *Results Analyzer* uses part of these data to (i) evaluate the volume vs performance and the volume vs emissions trade-off, and (ii) to rank the DQ dimensions according to their impact on the model performance. As an example, the results of these two analyses performed by the *Results Analyzer* for three

datasets and three models are shown and discussed here. In Fig. 4, the impact of data volume on CO₂ emissions (top row) can be compared to its impact on the model performance (bottom row). While the degradation in performance due to a reduced training set grows slowly, the CO₂ emissions have a steeper trend, suggesting that the gain in terms of environmental sustainability beats the loss in terms of performance. It can be seen that volume reduction has a limited effect on the second and third experiments. This can be due to the dataset characteristics (a better class separability, which makes it easier to build a high-performance model with fewer data) and its intertwining with the selected DL model and HP configuration. Figure 5 shows the impact of DQ degradation on the resulting model performance, considering three different DQ metrics: completeness, consistency, and accuracy. It can be observed that not all the metrics have the same impact, with completeness being the less relevant aspect. A ranking of the most relevant DQ dimensions is extracted from these experiments and stored in the *Smart Reduction* DB. The intuition is that removing poor-quality data improves the overall model performance. To validate this intuition, we executed some experiments showing the results of the model performance under three different conditions: given a dataset with a percentage p of poor quality data (i) no data are removed; (ii) all the poor quality data are removed (Smart Removal); (iii) the same percentage p of data is removed but with a random selection. The experiments tested different percentages of affected data points for different DQ metrics. Results are shown in Fig. 6. As can be observed, smart removal performs similarly or better than the other two options.

The *Reduction Curves Extractor* uses the experiments DB to build a set of reduction curves modeling the trade-off between performance and volume. To build the *Reduction Curves* DB, 42 datasets and three models were used. The reduction curves were modeled as shown in Eq. 4:

$$F1_Score = C_1 + C_2 \times \log(ds_p) \quad (4)$$

where ds_p is the percentage of the original dataset to be considered, C_1 and C_2 are the regression parameters, and $F1_Score$ is the resulting model performance.

5.2 Data Selection Implementation

The content of the *Reduction Curves* DB is used by the *Regression Model Generator* to build a Regression Model. In our implementation, we tested several algorithms and selected the Random Forest Regression [18]. All the details about the inputs and output of this model can be seen in Table 1.

As described in Sect. 4, our methodology considers a HITL approach. For this, it is expected for a researcher to provide all the necessary information (\mathcal{DT} , \mathcal{MT} , \mathcal{G}) and to perform a preliminary HP search process with a reduced dataset (\mathcal{BT}). In our tests, we set the dataset size $ds_p = 50\%$ since this value resulted in a good trade-off between performance and emissions in the analysed scenario. The *Dataset Feature Extractor* extract from the dataset the missing characteristics for the selected data type and model (as described in Table 1) and assesses DQ.

Table 1. Inputs of the regression model

Input Type	Attribute Name	Description
Model Metadata	Dataset Type	What is the source of the data (chosen from categories)
	Number of Classes	How many classes has the dataset
	Number of Training Samples	How many training samples are available in the full dataset
	Length of Sequence	How long is each time series
	Dimensions	How many dimensions are in each time series
Dataset Metadata	Architecture Type	Which is the general architecture type of the model (chosen from categories)
	Number of Parameters	How many parameters does the model have

Using this information, the Regression Model can be exploited to obtain the C_2 coefficient for the new regression curve. The C_1 coefficient is computed using the baseline $F1 - Score$ result from \mathcal{BT} , using Eq. 5. With this reduction curve and the performance goal \mathcal{G} , the required dataset percentage is computed by the *Reduction Estimator* using Eq. 6. Finally, the dataset is reduced by the *Dataset Reducer* component by removing low-quality data first according to the DQ dimensions ranking until the required percentage is met: (i) for completeness, data points containing null values are removed; (ii) for consistency, data points with the same values but different target values are removed. Since data points associated with a wrong label cannot be automatically detected, no action is taken to improve accuracy unless additional information is provided. The Data Selection phase additionally allows the researcher to express preferences on the class balance of the resulting dataset: the user can decide if to keep the same distribution or reduce as much as possible the imbalance between classes.

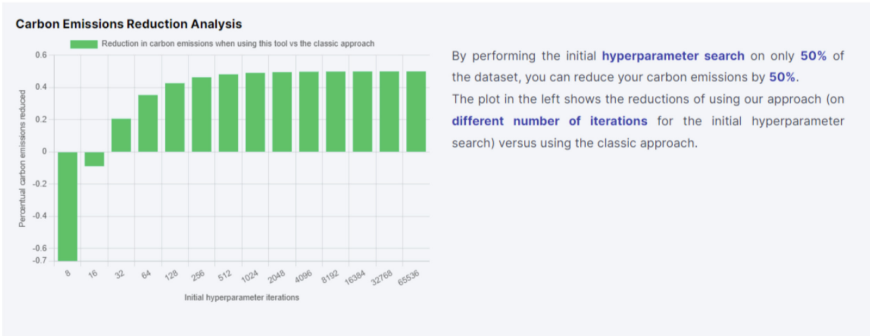
$$\hat{C}_1 = \text{ReportedF1Score} - \hat{C}_2 \times \log(ds_p) \quad (5)$$

$$\text{RequiredPercentage} = e^{\frac{\text{GoalMetric} - \hat{C}_1}{\hat{C}_2}} \quad (6)$$

To ease the interaction with the researcher, we provided a prototype including a web interface (Fig. 7a). To tool increases the sustainability awareness of the researcher by estimating the emissions reduction of the approach (Fig. 7b).



(a) User Interface of the tool



(b) Prediction of the reduction in CO₂ emissions

Fig. 7. The Data-Centric Green DL tool GUI

6 Validation

The validation of the proposed methodology needs to focus on two aspects: (i) using the approach, the environmental impact of DL model training is reduced; (ii) the performance goals set by the researchers are met.

The majority of carbon emissions produced in a DL pipeline come from the HP search. Using a classic method for this process, N training iterations are usually performed on the full dataset changing the HP values, and the resulting best model is chosen. This paper proposes to perform this search in two steps: (i) N training iterations are performed on a reduced dataset $ds_p = 50\%$ to generate the required input for the methodology; (ii) a final HP search is refined on the resulting reduced dataset $ds_{\hat{p}}$ with n final iterations. If $N > n$ by a significant

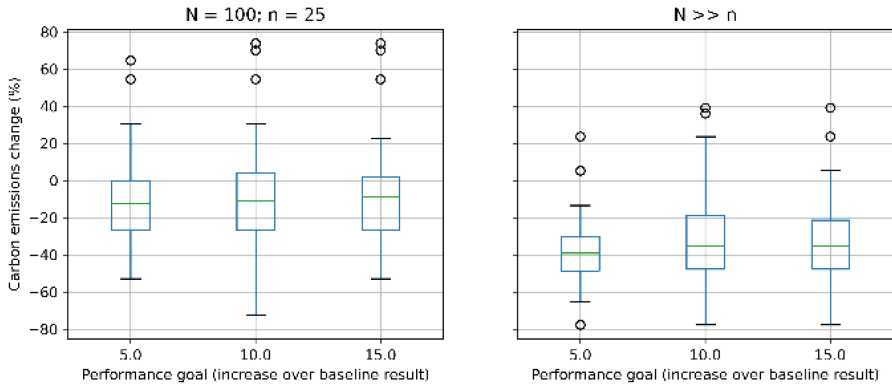


Fig. 8. Carbon emissions change due to the proposed approach

amount, the carbon emissions of the new method will be less than the ones in the classic method, up to half in the limit case where $N \gg n$. The values for N and n used for the experiments were defined part experimentally and part from the literature [1].

In order to extensively and systematically validate the approach, the same experimental data obtained in Sect. 5 were used. The data contained in the *Experiments* DB were split into a training set (70% of the experiments) for training the regression model and a testing set (30% of the experiments) to simulate new experiments requested by researchers. The baseline result \mathcal{BI} was obtained from one of the sub-experiments performed from the validation set, and the performance goal \mathcal{G} was set as the performance of the selected sub-experiment plus 5%, 10%, and 15%. Taking an extreme case, where $N \gg n$, the emissions were reduced by around 40%, on the three performance goal cases. When using more reasonable values of iterations for the HP search ($N = 100; n = 25$), the reduction in emissions was closer to 15% (Fig. 8). Figure 9 shows instead the average error of the approach of predicting the model performance for a specific dataset volume, which resulted to be near 1.5%.

Finally, an extra end-to-end experiment was performed, to test how the researcher can reduce carbon emissions on a new and unseen DL model. This was done using the Swedish Leaf dataset⁵, modified to have *consistency* = 0.85 and with a performance goal set as $\mathcal{G} : F1 - Score = 0.95$. After the first HP search with $N = 100$, a baseline result of $F1 - Score = 0.91$ was achieved. With the proposed approach, the original dataset was reduced using 68% of the original data, with a resulting consistency of 1. The new dataset was used to perform a second HP search with $n = 1$, with a resulting performance of $F1 - Score = 0.961$. Table 2 shows a comparison between the proposed method and the results of the

⁵ <http://www.timeseriesclassification.com/description.php?Dataset=SwedishLeaf>.

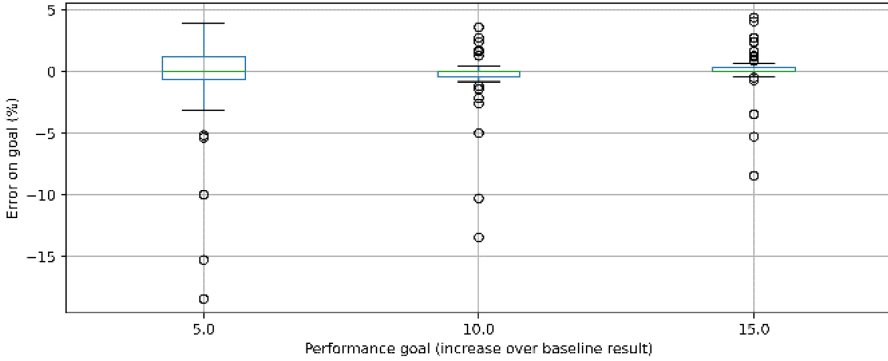


Fig. 9. Error in the satisfaction of the performance goal set by researcher

Table 2. Performance and emissions results of the Green DL compared with classic DL training on the Swedish Leaf dataset

Approach	Data Volume	Consistency	Emissions	F1-Score
Proposed	50%	1	0.0424 kg CO ₂ e	0.961
	68%	1		
Classic	100%	0.85	0.07451 kg CO ₂ e	0.91
Classic	85%	1	0.05654 kg CO ₂ e	0.944

classic method in two different cases: one where the full dataset was used (with the inconsistent series present), and one with all the inconsistent data removed (85% of the dataset). The proposed method generated fewer emissions when compared to both cases while reaching the performance goal set.

All the experiments executed in this paper generated 6.7kg CO₂e. Using the tool on datasets with a size similar to the ones used for development, with $N = 100$; $n = 25$, we can estimate that the generated emissions would be offset with 274 uses of the tool. This number is reduced to only 12 uses with $N = 1000$.

The preliminary results obtained in testing the approach have proven the relevance of data preparation for Green DL. However, the approach can be enriched by i) integrating it with the existing model-centric approaches, providing a holistic view to Green DL; ii) exploiting additional data features affecting either model performance or energy consumption (e.g., data augmentation and class balancing). Namely, the approach can also be applied to other DL tasks, however, additional experiments will be needed to check its efficiency and to provide a way to automatize the parameters optimization in different scenarios.

7 Conclusion

Motivated by the increasing environmental impact that DL is having, this paper proposes a data-centric approach for reducing carbon emissions in DL training

pipelines as part of what is called “Green AI”. This research is data-centric since all the considerations to reduce energy usage are addressed to more efficient use of the training data, rather than focusing on more efficient hardware or algorithms. For this, characteristics like data volume and DQ were taken into account. A general methodology, valid for any DL task, was proposed, consisting of two phases. First, a Data Exploration Phase inspected the characteristics of the data and generated a knowledge base for efficient data reduction. Second, a Reduction Building System Phase is defined to support researchers in reducing their carbon emissions by operating on the training dataset. This process follows a HITL approach, where the researcher needs to interact with it, providing all the necessary information.

An implementation of the approach focusing on the time series classification task using DNNs is provided. The result of the implementation is a prototype that can be used by the researchers. Experimental results showed that the approach can reduce carbon emissions by up to 50%. With time, more data from experiments of new model architectures and datasets can be included, further increasing the accuracy of the predictions provided by the proposed system.

Future work will focus on testing a more extensive set of data-centric characteristics, to reduce more or in a better way the dataset. Also, the proposed system could be integrated with a location-aware deployment service, which can train models in locations with a more favourable energy mix.

Acknowledgements. This research was supported by the EU Horizon Framework grant agreement 101070186 (TEADAL) and by the Spoke 1 “FutureHPC & BigData” of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU).

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(2), 281–305 (2012)
2. Berti-Equille, L.: Learn2Clean: optimizing the sequence of tasks for web data preparation. In: *The World Wide Web Conference*, pp. 2580–2586 (2019)
3. Budach, L., et al.: The effects of data quality on machine learning performance. preprint [arXiv:2207.14529](https://arxiv.org/abs/2207.14529) (2022)
4. Castanyer, R.C., Martínez-Fernández, S., Franch, X.: Which design decisions in AI-enabled mobile applications contribute to greener AI? preprint [arXiv:2109.15284](https://arxiv.org/abs/2109.15284) (2021)
5. Dong, X.L., Srivastava, D.: Big data integration. In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 1245–1248. IEEE (2013)
6. Frey, N.C., et al.: Energy-aware neural architecture selection and hyperparameter optimization. In: *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 732–741. IEEE (2022)
7. He, K., et al.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
8. Hsiao, T.Y., et al.: Filter-based deep-compression with global average pooling for convolutional networks. *J. Syst. Archit.* **95**, 9–18 (2019)

9. Jain, A., et al.: Overview and importance of data quality for machine learning tasks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3561–3562 (2020)
10. Knight, W.: AI can do great things - if it doesn't burn the planet. *Wired Magazine* (2020)
11. Konstantinou, N., Paton, N.W.: Feedback driven improvement of data preparation pipelines. *Inf. Syst.* **92**, 101480 (2020)
12. Lucivero, F.: Big data, big waste? A reflection on the environmental sustainability of big data initiatives. *Sci. Eng. Ethics* **26**(2), 1009–1030 (2020). <https://doi.org/10.1007/s11948-019-00171-7>
13. Maccioni, A., Torlone, R.: KAYAK: a framework for just-in-time data preparation in a data lake. In: Krogstie, J., Reijers, H.A. (eds.) CAiSE 2018. LNCS, vol. 10816, pp. 474–489. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91563-0_29
14. Miao, Z., et al.: A data preparation framework for cleaning electronic health records and assessing cleaning outcomes for secondary analysis. *Inf. Syst.* **111**, 102130 (2023)
15. Patterson, D., et al.: Carbon emissions and large neural network training. preprint [arXiv:2104.10350](https://arxiv.org/abs/2104.10350) (2021)
16. Rolnick, D., et al.: Tackling climate change with machine learning. *ACM Comput. Surv. (CSUR)* **55**(2), 1–96 (2022)
17. Schwartz, R., et al.: Green AI. *Commun. ACM* **63**(12), 54–63 (2020)
18. Segal, M.R.: Machine learning benchmarks and random forest regression. UCSF: Center for Bioinformatics and Molecular Biostatistics (2004)
19. Shin, Y., et al.: Practical methods of image data preprocessing for enhancing the performance of deep learning based road crack detection. *ICIC Express Lett. Part B Appl.* **11**(4), 373–379 (2020)
20. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. preprint [arXiv:1906.02243](https://arxiv.org/abs/1906.02243), June 2019
21. Sun, C., et al.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 843–852 (2017)
22. Werner de Vargas, V., et al.: Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. *Knowl. Inf. Syst.* **65**(1), 31–57 (2023). <https://doi.org/10.1007/s10115-022-01772-8>
23. Verdecchia, R., et al.: Data-centric green AI: an exploratory empirical study. preprint [arXiv:2204.02766](https://arxiv.org/abs/2204.02766) (2022)
24. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1578–1585. IEEE (2017)
25. Xu, J., et al.: A survey on green deep learning. preprint [arXiv:2111.05193](https://arxiv.org/abs/2111.05193) (2021)