École Polytechnique Fédérale de Lausanne

CUMULATOR — a tool to quantify and report the carbon footprint of machine learning computations and communication in academia and healthcare

by Tristan Trébaol

Semester Project Report

Approved by the Examining Committee:

Mary-Anne Hartley, Hossein Shokri Ghadikolaei
Project Advisor

Martin Jaggi
Project Supervisor

EPFL IC IINFCOM MLO INJ 341 (Bâtiment INJ)
Station 14
CH-1015 Lausanne

June 26, 2020

Abstract

"The cost of training machines is becoming a problem". This is the title of an article from The Economist published in June 2020 that highlights the staggeringly unappreciated the financial impact of AI [38]. However, these costs on are not limited to monetary concerns but also accrue a concerning environmental toll. One round of training for some of the most complex machine learning models can emit millions of kilograms of carbon dioxide due to the electricity consumed*. With the growing popularity of ML and the digitization across all sectors, there should be a growing awareness of the potential impact of these technologies on the environment and their potential contribution to climate change. Then, their use could be rationalized and steps can be taken to responsibilize the offset of their impact. We hereby propose *CUMULATOR*: an open-source API to quantify and report the carbon footprint of machine learning methods. As a demonstration, we integrated this API within an ML based medical research platform called *Alg-E*, which will be used in a large scale medical research project in Tanzania and Rwanda. We use *CUMULATOR* to analyse the trade-off between accuracy and carbon footprint within *Alg-E* and extend it with simple visualisations. In parallel, we also propose a *Carbon Statement Protocol* to quantify and report the carbon footprint of individual work, which uses this project as a proof-of-concept. This protocol and *CUMULATOR* thus comprise a great set of tools to report the carbon footprint of a large-scale medical research trial and EPFL research projects in the future.

*The following syllogism:
First premise: *«one round of training for the biggest models can cost "millions of dollars" in electricity consumption »*. This sentence is an excerpt from an article by *The Economist* that reports the words of Jerome Pesenti, Facebook head of AI, in June 2020 [38].
Second premise: *one million of $US dollars in electricity consumption is roughly equivalent to the emissions of 4.8 millions kilograms of CO_2 along the electricity life cycle.* Indeed, on average in the US, the electricity cost is 10.29 cents/kWh ([42], in 2020) and 494 grams of carbon dioxide is emitted per kilowatt-hour consumed ([18], in 2018). This makes 4.8 kgCO_2/$US by simple unit conversions.
Conclusion: *one round of training for the biggest [machine learning] models can emit millions of kilograms of carbon dioxide due to the electricity consumed.*

Acknowledgments

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

— Carl Friedrich Gauss

I am thankful to Annie, Martin, and Hossein for their continued enthusiasm and support during the project. I am grateful to Annie for bringing constructive suggestions and creative ideas in our weekly virtual meetups. Despite the COVID crisis, I felt that she was always around. There is no better motivation than to read her celebrating interjections answering my late Slack messages. Also a particular thanks to Martin for listening to my questions and guiding me during my master at EPFL. I was systematically surprised by his immediate availability each time I came to his office with random topics to discuss. I wish to thank Hossein who provided me with truly interesting literature full of personalised comments in the first weeks. I enjoyed the freedom I had to choose a direction for this project and explore the tracks I found more interesting.

A really sincere thanks to Liamarcia Bifano for the fun and productive code pairing sessions across the world, her insightful explanations, and programming tips, and for the excellent quote. All the new features added to Alg-E platform was thanks to her support. I shall remember her capacity to celebrate each little achievement as an immense victory.

Finally, I wish to thank all the members of the Machine Learning and Optimization Laboratory of EPFL for their warm welcome, positive remarks about the outcome of the project, and for showing their support to the logo of *CUMULATOR*. A special appreciation to Felix Grimberg for his participation to advocate the uniqueness of the word cumulator on the internet.

Lausanne, June 26, 2020 Tristan Trébaol

Background

Climate Change is a complex phenomenon that involves global temperatures and sea-level rise, warming and acidification of oceans, glacial retreat, decreasing snow cover, extreme events on earth. The Intergovernmental Panel on Climate Change reported that "Scientific evidence for warming of the climate system is unequivocal" [9]. Its evolution is closely connected with many factors which influence energy flows in earth atmosphere, including greenhouse gases (mainly water vapor H_2O, carbon dioxide CO_2, methane CH_4, nitrous oxide N_2O, ozone O_3). Those molecules are mostly generated by a static set of sectors (Electricity & Heat production 25%; Agriculture, Forestry and Other Land Use 24%, Industry 21%; Transportation 14%; Buildings 6%; Other 10% [13]) and their emissions is dramatically rising (Figure 1.1). The carbon footprint measured in mass of carbone dioxide equivalent (gCO_2eq) is indeed a standard indicator to quantify the amount of greenhouse gas emissions [6] [28]. In a nutshell, the world energy resources are limited and electricity production has the highest impact on climate change.

Moreover, subdomains such as the IT sector are rapidly evolving, and avidly consume energy. According to *the Shift Project*, a french think-tank about digital sobriety, the energy consumption of digital technologies will double between 2015 and 2025 ([24], "Full Report"). While images of landfills with piles of disused computers can help convey the cost of wasteful hardware manufacture and use, the carbon cost of computing is much less tangible. Indeed a single Google search consumes on average 0.2 gCO_2eq [17]. As each second 63,000 Google searches are launched, this has an equivalent footprint of 12.6 $kgCO_2eq$ per second on any given day [11].

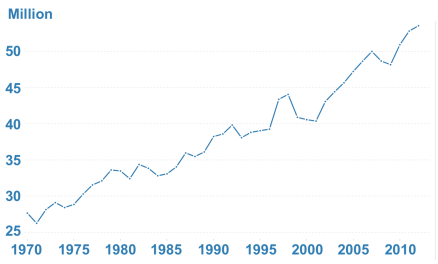


Figure 1.1: Evolution of total greenhouse gas emissions by human activities (kilotons of CO_2eq), 1970-2012 [41]

Perhaps surprisingly, there is currently little research in sustainable AI and only a few tools to monitor the carbon footprint of computing with no independent benchmarking. Indeed, most AI papers tend to target accuracy rather than efficiency [35]. Several research groups called for such a reference tool to estimate carbon footprint of machine learning (ML) algorithms [23] [37] [16]. The community can be optimistic since frameworks to compare ML methods have already been successfully implemented. For instance, MLBench is an attempt to provide a fair benchmarking suite for referencing distributed ML methods [29]. Also, the expansion of AI is massive. In the US, the share of jobs in AI-related topics increased from 0.26% of total jobs posted in 2010 to 1.32% in October 2019, with the highest share in machine learning (0.51% of total jobs). AI labor demand is growing especially in high-tech services and the manufacturing sector [4]. The number of published AI-related papers grew from 10k in 2000 to 60k in 2017 [5]. Due to the long time needed to fully deploy a new technology to industry, one can expect that this increase in AI-related research will have a long

term impact on the industry job market. Also, complex ML models are heavy energy consumers. In 2019, researchers at the University of Massachusetts at Amherst released a startling report estimating that the amount of power required for training and searching a certain neural network architecture involves the emissions of roughly 284 tons of carbon dioxide, which is equivalent to 177 roundtrip flights between Paris and New York [37] [26].

Hence, from the dependency of machine learning on energy consumption and thus its impact on climate change, the broader the usage of large scale machine learning methods, the more critical becomes its optimisation in terms of carbon footprint. We should, therefore, consider computing as a limited resource, and its use should be optimised and equitably distributed both in its impact and target population.

AI innovations are going to revolutionise and democratise equitable access to health care in resource-limited settings, including for diagnosis, patient management, and the rise of continuous distributed wearable health monitoring. Certainly, this is a worthwhile investment on which to spend carbon, but optimised methods would nevertheless be able to minimise its environmental impact. In the wake of genetic testing the size and dimensionality of datasets are gargantuan (the European Molecular Biology and Bioinformatics Institute Laboratory hosts 14PB of data and the numbers are expected to double every year, GenBank hosts over 120 billion DNA bases) [25]. They include millions of features as well as large quantity data in the form of MRI images, microscopy, and text reports. Focusing on healthcare in limited-resource settings, a tool collecting medical data (called e-POCT, for "electronic point of care test") is being trialed in several countries in Africa with plans for a larger scale up over the coming years. It guides clinicians through the entire consultation and recommends treatment based on a few clinical signs and POCT results [21]. Several aspects of this tool are being improved with ML models, through the development of the *Alg-E* website, to provide more accurate and adaptive symptom-to-treatment support while keeping interpretability and transparency.

This project investigates how to report the carbon footprint of ML methods, with application on the *Alg-E* website and academia. As a proof of concept, we will report the carbon footprint of this project and present a protocol for future researchers to follow to also derive the environmental impact of their research.

Aim and objectives

This project aims to raise awareness about the carbon footprint of ML methods and to encourage further optimisation and the rationale use of AI-powered tools. For this, the project will focus on communicating the carbon footprint of two applications 1) in a large scale medical data analysis platform, and 2) in academia, to build an early awareness among clinicians, students and researchers about the importance of the environmental impact of digital technologies.

General objective

Objective 1, Perform a landscape analysis of the existing tools for calculating the carbon footprint of computing and benchmark them

Objective 2, Present *CUMULATOR* which will be used to assess the carbon footprint of ML methods in this project

Medical data analysis

Objective 3, Integrate *CUMULATOR* to the *Alg-E* platform, to compute the carbon footprint of each model under various conditions and communicate the trade-off between accuracy and carbon footprint on the user interface

Objective 4, Generate a list of features required to calculate the carbon footprint of a large-scale medical data analysis platform, such as will be used in the *DYNAMIC* trial

Academic data analysis

Objective 5, Create a protocol to help future researchers and students quantify the carbon footprint of their projects

Contents

Anthropology of the carbon footprint in ML methods

This chapter provides the relevant material to support the project method, results, and limitations. As mentioned in the background, the environmental impact of ML methods is directly linked with its energy consumption. Carbon footprint can be derived by 1) calculating the energy consumption of ML systems (computations and communication) and 2) converting this energy consumption into an estimation of the carbon footprint (using the geographical distribution of carbon footprint per unit of energy produced), see Figure 3.1.

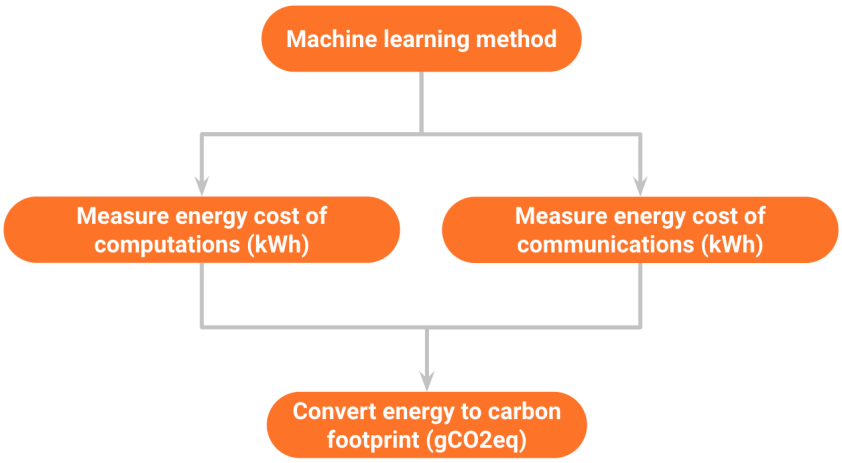


Figure 3.1: Process to report the carbon footprint of ML methods

From an energy-meter perspective, ML methods consume energy by two different means. The computations which happen locally inside a processor and the communication of data packets before (dataset generation), during (parameter updates between different nodes of a ML network) or after (model deployment) the training of the model. The rest of this section presents the different concepts involved in the evaluation of the carbon footprint.

3.1 Computational costs

The energy consumption of computations is reported usually either from 1) time of computation, 2) GPU/CPU (i.e. Graphic/Central Processing Unit) utilisation as a percentage, 3) FLOPs (i.e. floating point operations per second) related metrics, 4) the number of parameters of the model 5) an energy monitoring interfaces, Running Average Power Limit (RAPL) or Application Power Management (APM) [16]. However, an ongoing study showed that both FLOPs metrics and the number of parameters of the model correlate poorly with energy consumption ([16] Appendix B,E). Also, the GPU/CPU utilisation as a percentage refers to the number of busy clock cycles over the number of clock cycles ran. The difficulty of the task (simple or expensive computations) is not taken into account, and hence it is difficult to base precise energy consumption on this criteria alone. The rest of this section describes the two remaining methods (1 and 5).

3.1.1 Time-based energy consumption

The calculation of the energy consumption base on time is straightforward: multiply the duration of the computations (recorded at run time) by the Thermal Design Power (TDP) of the processor (which is the power consumption under the maximum theoretical load [39]) and by the number of processor(s) used. Aside from the wall clock training time, the duration can also be derived from the time to run one full forward pass (and information on the number of steps before convergence is needed to compute total time), which enables us to differentiate the computation and the communication costs. The precision of the energy consumption can be improved by referencing the right processor's power consumption.

The main limitation of this method is that it assumes a 100% of GPU utilisation at maximum load, hereby disregarding the optimisation made by frameworks used at a lower level to simplify the complexity of computation load [16].

3.1.2 Energy and power monitoring at run time

In physics, power is the amount of energy transferred or converted per unit time. It is indeed measured in the International System of Units in wattage which is a joule (a measure of energy) per second (a measure of time).

As large scale ML tasks often require heavy computations and equally heavy energy consumption, the more time and energy-efficient the processors are, the better. Power consumption is therefore a key indicator of attractiveness for processor manufacturers. However, proving power efficiency necessarily relies on the ability to perform power measurements at some point. Even though ammeters are usually used for this, direct access to the component is needed which is often difficult and not implemented as a base feature in modern complex miniaturised systems.

To simplify power measurements at a CPU/GPU level, recent processors support software-based power estimators such as RAPL and APM respectively produced by Intel and Advanced Micro Devices (AMD) which have a duopoly in CPU shares worldwide, [2].

RAPL (Running Average Power Limit) and APM (Application Power Management) are low-level interfaces based on a set of *hardware performance counters* (special-purpose registers built into modern microprocessors to store the counts of hardware-related activities within computer systems) which provide energy consumption information. In particular, they use *model-specific registers* (MSR - control registers in the x86 instruction set used for debugging, program execution tracing, computer performance monitoring, and toggling certain CPU features). For a complete understanding, it is also worth to define here what is a *processor register*: a quickly accessible location available to a computer's central processing unit (CPU). The main purpose of those models is to provide energy and power monitoring at run-time. This makes them excellent candidates for carbon footprint estimation.

Even though these models are widely used, they are subject to limitations:

- They remain predictors of energy consumption, and do not provide physical measurements.
- The update criteria for each new measurement on RAPL and APM are implemented differently. Hence, for studies at time scales lower than a second, one should parametrise those models carefully to have significant results, [14] section III.D
- Several studies were performed to evaluate the accuracy of RAPL and APM compared to state of the art power meters. It found that the models have a systematic error [14], and an accuracy error of up to 22% [33] [22]
- A 2015 study benchmark of RAPL (versus high accuracy power meter) for different Intel processors micro-architectures revealed an accuracy increase on newer Intel processors. Indeed the RAPL implementation on the Haswell micro-architecture (2013) was almost perfectly correlated with the power measurements [15] [22]

Energy monitoring with model-based interfaces is a working solution for environmental impact calculations.

3.2 Communication costs

In recent years, data centers have emerged as an essential back-end infrastructure of the Information and Communications Technology (ICT) industry [3]. ML becomes heavily used for

large scale applications in the cloud [20], where the data is spatially distributed and needs to stay local for privacy and energy preserving reasons. Recent research on optimisation of WAN communication in geo-distributed ML led to a x1.8 to x53 training speedup with zero side-effects [19]. This suggests that there is a large margin for the optimisation of communications in state of the art methods.

In communications, the types of network (wired or wireless, LAN or WAN, etc) and the number of relay nodes influence the energy consumption to transmit one data packet. Hence, it is complex to compute the environmental impact. Knowing that large scale ML applications are mostly cloud-based, we introduce work done by *the Shift Project* to evaluate the environmental impact of data traffic in data centers ([24], "Materials", "Forecast Model"). Here the *global data traffic* also referred to as *data center IP traffic* includes communications from 1) data center to a user, 2) within data centers and 3) between data centers. In order to derive the carbon footprint of communications, the interesting value is the **energy consumption per-byte of data transferred**, as shown in Figure 3.2. We obtained it by dividing the energy consumption of data centers (which scales in 100TWh) with the global data traffic (which scales in ZB, i.e. zettabytes, or 10^{21} bytes). Their 2018 analysis reports those values between 2013 and 2016 and provides estimates for 2017-2025 based on annual reports of the International Data Corporation, Cisco, and Gartner [8].

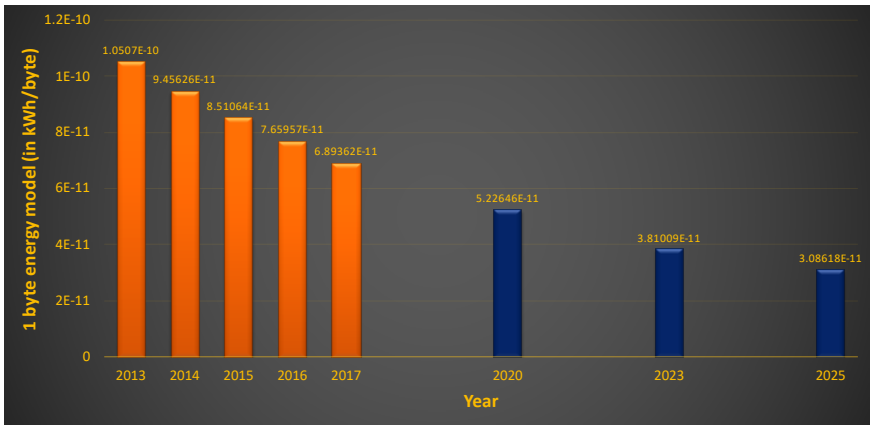


Figure 3.2: Evolution and prediction of the "1byte model", i.e. energy consumed per-byte of data traffic in data centers

Data centers are becoming energy efficient with time: in Figure 3.2 the per-byte energy cost could be divided by two between 2017 and 2025. The carbon footprint due to communication in ML methods is directly influenced by this trend.

3.3 Energy to carbon footprint conversion

The last step to derive the carbon footprint is to convert the energy consumption thanks to the **carbon intensity**. The latter is defined as the amount of greenhouse gas emissions in grams carbon dioxide equivalent per unit of energy produced (gCO_2eq/kWh). This value depends mainly on type of energy source (Figure 3.3) used to provide the electricity grid at the location of the computing machine or data center. To our knowledge, the most complete map of carbon intensities was created by an open source project called *Electricity Map*. Energy resources are unequally distributed on earth, which leads to high variance of carbon intensity depending on the geographical region (France: 68 gCO_2/kWh, UK: 159, the Netherlands: 454, Australia: 492-761 depending on the region [43]) . However, the carbon intensity varies not only on the source of energy, but also on the losses along the electric pathway. Indeed, the average European carbon intensity increases by 130% between production and consumption [31]. Thus, the quality of the electricity distribution infrastructure impacts the carbon intensity at consumption. In conclusion, the most important variable to report for this energy to carbon footprint conversion is the geographic location at which the computations are made.

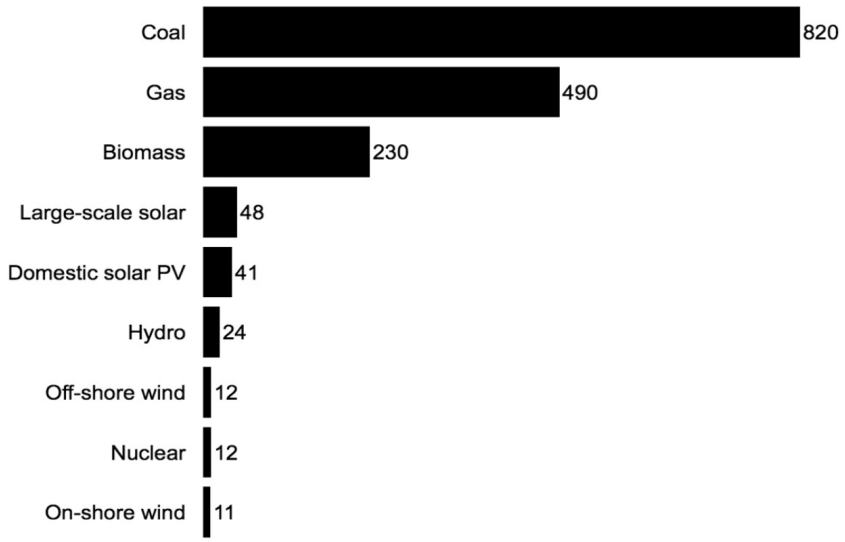


Figure 3.3: Life cycle emissions from electricity generation in gCO_2eq/kWh [7]

Since ML is mostly on the cloud, we investigate material to assess the carbon intensities of data centers. According to Google patent "US20140247537A1", about the *Medium Voltage Power Distribution in Data Centers*, data centers are connected to 4160 VAC (which is voltage delivered in alternative current) on the grid and perform a step-down conversion to 240 VAC on-site. This may not be the case for all data centers. However, it is reasonable to assume that the losses during step down from High Voltage (HV) to Low Voltage (LV) will be of the same order of magnitude when done on the grid or on-site. Since GPUs work on standard voltages (220-240 V)

we should refer to the carbon intensity consumed at LV. The international standard "IEC 60038" indeed describes voltages below 1 kV as Low Voltages.

Method

This chapter presents and justifies the chosen approach for each objective.

4.1 Objective 1, Perform a landscape analysis of the existing tools for calculating the carbon footprint of computing and benchmark them

Since little previous work was undertaken at the EPFL Machine Learning and Optimization Laboratory (MLO) regarding the carbon footprint of ML methods, a preliminary analysis of the state of the art on the subject was necessary. We used a systematic approach to gather as much information as possible. The method was to search on the web, on online research papers agglomerators (sci-hub, google scholar), and on websites for developer community (GitHub, PyPI) with the keywords linked to the project, namely machine learning, calculation, computation, carbon footprint, environmental impact, sustainable AI, tracker, API.

4.2 Objective 2, Present *CUMULATOR* which will be used to assess the carbon footprint of ML methods in this project

There are two high-level use cases of *CUMULATOR*. In research, it could be used to bring an environmental aspect in the optimisation of ML methods at different levels (multi-node ML algorithms, ML model architecture and parametrisation, etc). In research and real-world applications, it could be used to report the carbon footprint to raise awareness about the environmental impact of ML.

Below is a list of the important **project requirements** on which we based our trade-off between developing our API, and using or extending an already existing one. The results of the landscape analysis will then be confronted with those requirements to make the decision.

1. The API can measure or calculate the energy consumption of computation and communication of ML models and convert them into carbon footprint. This requirement is related to the material introduced in 3.
2. **High compatibility**: the API shall rely on open-source packages and knowledge to be usable under a large number of circumstances. In the literature one observes a trade-off between precision of the carbon footprint estimations and compatibility criterion (for example RAPL and APM introduced in 3.1 requires special GPU specifications). Even though we choose to place the importance on a wide operating range, the API could also have restrictive modes that provide more accurate estimations.
3. **Simple integration within ML projects**: few commands should be necessary to integrate automatic computation and reporting of the carbon footprint of ML computation and communication in ML projects, such as inside the *Alg-E* platform presented in 4.3. (open source library, python, have a static approach)
4. **Open to contributions**: the API shall be accessible future students and researchers to extend it with other features or existing APIs. This is a critical requirement for the perpetuity of the project.
5. **Verifiability**: the API shall rely on open-source data and the calculations should be explained.

4.3 Objective 3, Integrate *CUMULATOR* to the *Alg-E* platform, to compute the carbon footprint of each model under various conditions and communicate the trade-off between accuracy and carbon footprint on the user interface

Extending *Alg-E* with *CUMULATOR* is a small step for knowledge in feature integration but a substantial step towards raising awareness about the environmental impact of machine learning. Since *Alg-E* is still in the development phase, the impact of this analysis can be suggestions to improve the platform (model accuracy, interpretability, etc). After deployment in the *DYNAMIC* trial, *Alg-E* will display the carbon footprint next to the other performance metrics. This will inform the clinicians about the carbon footprint of the algorithms they ran. Adding such information could impact the behavior of the clinicians. For example, if inequalities in the carbon cost of the models are observable, the clinicians could adapt their behavior by avoiding to use a certain model, for better or for worse. The conclusions of this analysis should, therefore, be carefully considered.

It is also important to mention that *Alg-E* was developed by one main contributor in full-time

for six months. It is a remarkably well-finished product. However, the models implemented were not tested, hence this project also serves to evaluate the performance of the models.

4.3.1 The *Alg-E* platform

As part of the *DYNAMIC* trial, a machine learning platform to generate interpretable medically approved, data-responsive clinical decision algorithms was created in 2020 [1]. The platform allows clinicians to load tables, run ML models for classification or regression tasks, and analyse and interpret the results. Since the clinicians might not have a machine learning background, clarity, and interpretability of the user interface are of predominant importance. The user interface on Figure 4.1 is described in detail in [1], section 4.2.1.

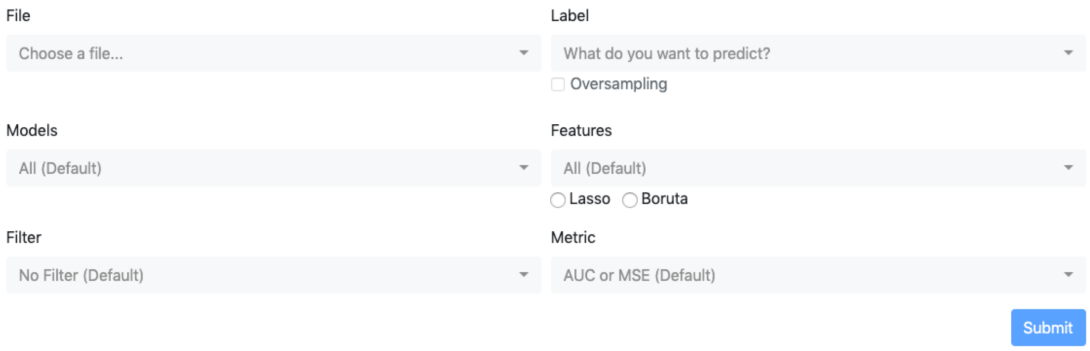


Figure 4.1: GUI of *Alg-E*'s menu to run classification (left) and regression (right) models

4.3.2 Definition of the study

The analysis of *Alg-E* with *CUMULATOR* is based on the observation of the performance metrics to 1) benchmark the models efficiencies and 2) investigate the impacts of the user-defined parameters (feature selection, oversampling). We summarise this in the four objectives below:

1. Examine if there is a cost difference between the models
2. Evaluate the capability of the models to fit complex datasets
3. Observe the benefit of feature selection methods for regression models
4. Study the impact of oversampling on classification tasks

4.3.3 Dataset selection and creation

To give a comprehensive overview of how the ML models used in *Alg-E* behave for the different sets of inputs and datasets, we decided to select datasets with a high number of samples (over 100k) and high dimensionality datasets (over 100 attributes). Three of the four datasets used in this study come from the UCI ML Repository and do not contain any missing values.

Dataset: ePOCT

The *ePOCT Data Set* was derived from a randomised controlled trial on 3'192 pediatric outpatients (aged 2 months to 5 years) which collected a large number of, mostly binary, features (e.g. presence/absence of a symptom) according to a rule-based decision tree. Thus, there are a large number of systematically missing values, as groups of questions are asked according to triage criteria. The dataset used here contains only 41 attributes and is aimed to test the *Alg-E* system on a range of classification and regression tasks. This is referred to as the simple dataset to perform the base tests.

Attributes description	**Type**
1: Age	Numerical
2: Unknown	Categorical
3-4: Final diagnosis (typhoid or severe anemia)	Categorical
5-8: Unknown	Categorical
9-25: Laboratory findings	Categorical and numerical
26-37: Signs	Numerical
38-47: Symptoms	Categorical

Dataset: CT images

The *Relative location of CT slices on axial axis Data Set* contains 385 attributes and 53'500 data points split among 96 patients. It is used for regression tasks.

Attributes description	**Type**
1: Patient ID	Numerical
2 - 241: Histogram describing bone structures	Numerical
242 - 385: Histogram describing air inclusions	Numerical
386 reference: Relative location of the image on the axial axis (class value). Values are in the range [0; 180] where 0 denotes the top of the head and 180 the soles of the feet.	Numerical

Dataset: Standing Postures

The *Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set* contains 561 attributes and 10'929 data points, which are records from accelerometer and gyroscope while the patient was performing 1 in 12 tested activities (e.g. walking, laying, standing, sitting and transitions between them).

Attributes description	**Type**
1: Patient ID	Numerical
2 - 560: Filtered values from 3-axis accelerometer and gyroscope acquired at 50 Hz, in time and frequency domains (using a Fast Fourier Transform)	Numerical
561: Related activity label (1-12)	Categorical

We use this dataset to test the efficiency of oversampling on a binary classification problem. We separated the activities between `Standing` and `Non-standing`. The activity `Standing` was chosen since it has the highest value counts (1929 in 10'929 samples), thus leading to a heavily unbalanced dataset as shown in Figure 4.2.

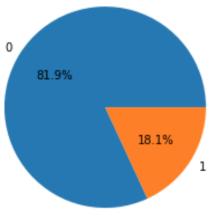


Figure 4.2: Distribution of Standing activities (label 1) versus Others (label 0) in the Standing Posture dataset

Dataset: Fonts

The *Character Font Images Data Set* consists of images from 153 character fonts with 411 attributes. The total number of images is 745'000.

Attributes description	**Type**
1: Font family	Categorical
2-12: Attributes about the format of the character	Categorical and Numerical
12-411: Pixel color from 0 to 255	Numerical

From this dataset, we created a classification problem by selecting the two fonts with the highest number of samples. The total number of samples is raised to 173'583, with 400 attributes (patient ID is not taken into account).

Font name	# samples
OCR-B (developed in 1968 for Monotype to facilitate Optical Character Recognition operations financial applications)	93'688
Segoe (the base font family used by Microsoft, also created by Monotype)	79'895

4.4 Objective 4, Generate a list of features required to calculate the carbon footprint of a large-scale medical data analysis platform, such as will be used in the *DYNAMIC* trial

The *DYNAMIC* trial aims to implement adaptable decision trees to guide the clinical management of pediatric patients in resource-limited settings. A "static" rule-based version of this tool was previously implemented in Tanzania in 2016-18, where it was able to reduce unnecessary antibiotic prescription and clinical failure. While these rule-based algorithms have had great success, their genericism blunts their accuracy in dynamic environments. For instance, the prevalence of many diseases varies over time (examples: flu generally occur in winter, poliovirus has been almost eradicated from Africa since 2016 [40]) and location (example: in 2018, 96 new disease outbreaks were reported across 36 of the 47 African Member States [27]). One goal of *DYNAMIC* is to use machine learning to adapt these generic, outdated decision tree depending on local updatable data that is relevant to surrounding epidemiology and available resources.

Looking at the data workflow of *DYNAMIC* in Figure 4.3, we can provide the required features to calculate the carbon footprint at each step of the data workflow during operations.

4.5 Objective 5, Create a protocol to help future researchers and students quantify the carbon footprint of their projects

For most of us, greenhouse gas emissions is a concept. We know that it is a significant issue, however, when it comes to numbers, it is difficult to quantify the seriousness of the situation without a reference value on which to base a comparison. This report aims to create some reference values. Being aware of **your carbon footprint** is a step towards is a step toward being able to make environmentally conscious choices. We suggest to standardise the reporting of carbon dioxide equivalent emissions of research projects by making a Carbon Impact Statement at the end of reports. Within *CUMULATOR* we will provide a script to determine the carbon

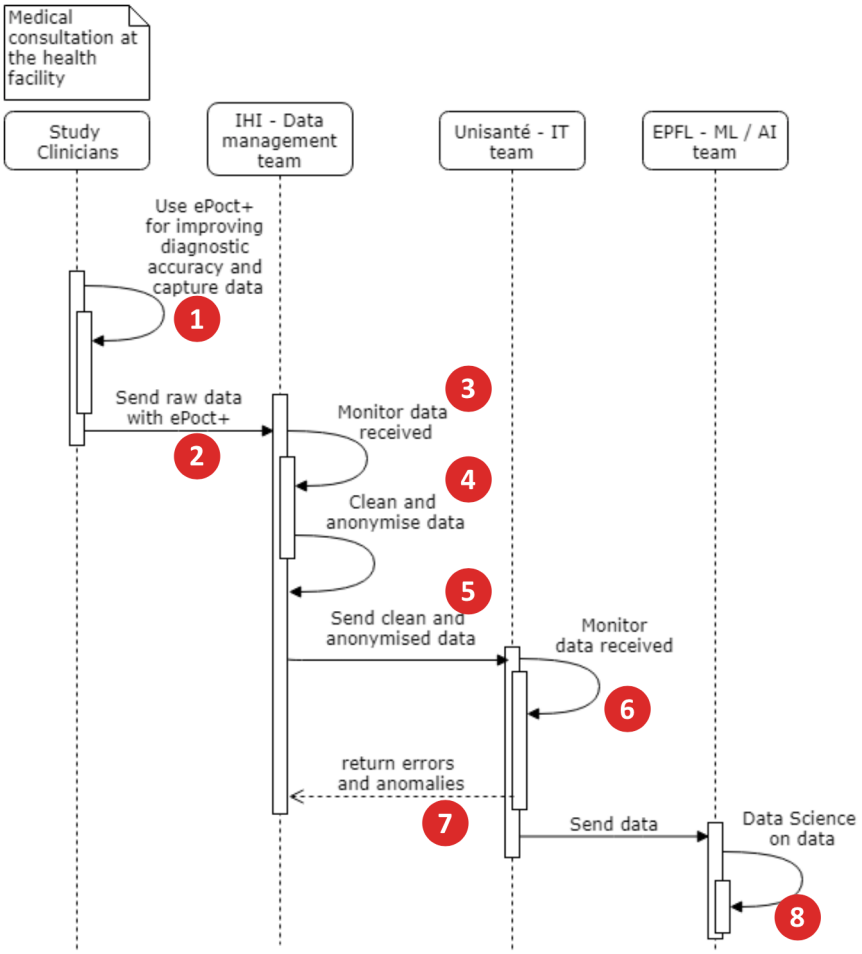


Figure 4.3: Workflow data of *DYNAMIC*

footprint for EPFL projects. Finally, we will use this project as an experiment to set up the most effective protocol to reduce the reporting-time overhead while ensuring that the reported value has a physical sense.

Results

5.1 Objective 1, Perform a landscape analysis of the existing tools for calculating the carbon footprint of computing and benchmark them

We made a comprehensive overview of the different tools available to analyse the environmental impacts of digital technologies. It is worth noting that most of the open-source so-called carbon trackers lack high-quality content, or focus on the costs linked to transportation, energy infrastructure, and concern individual or industry applications. Only a few tools, listed below, were created within an in-depth study around the topic of energy and carbon footprint of digital technologies:

- *The Shift Project* created ***Carbonalyser*** in 2018: it is a powerful add-on which analyses all the data traffic through your Internet browser (Google search query, webpage or image display, videos watched, files downloaded, etc). It accumulates all data automatically and reports it in real-time 1) the browsing time, 2) the amount of data downloaded, 3) the amount of electricity consumed 4) the carbon footprint. It is available for Chrome and Firefox web browsers, and there is a mobile version on Google Play.
- ***ECOFEN*** for Energy Consumption mOdel For End-to-end Network was created in 2011. *ECOFEN* allows us to analyse the energy profile of wired network equipment over time on top of the NS-3 network simulation software. It provides a precision of the order of milliwatts for the power and milliseconds for the time. This is however achieved at the price of intensive computations: for a large network topology (over 1000 nodes in 9 data centers) a 5 hour simulation is needed to capture one minute of network activity [10]. **Decision**. Even though this tool could provide very precise communication costs, we decided that it was not attractive enough to use in our study. Indeed, we would lose the simplicity of integration and would also require a large amount of effort to learn how to use it. However, it could be valuable to study some typical communication patterns associated with ML models to provide better cost estimates than the per-byte energy model from *the Shift Project* which we use in *CUMULATOR*.
- **The *ML Emissions Calculator*** is a time-based tracker (see 3.1) which was created in 2019 [23]. The carbon footprint is evaluated after the computations by entering parameters in their webpage. The parameters required are: the hardware type, hours used, cloud provider,

and region of compute. This calculator uses publicly available data for 4 main variables which impact the carbon footprint: 1) the energy consumption of hardware, 2) the location of providers' region of computation – which was assumed to be connected to their local grid, 3) the region's CO_2eq emissions per kWh and 4) potential offsets bought by the provider.

- **The *Experiment Impact Tracker*** is a simple drop-in method to track energy usage, carbon emissions, and compute utilisation of computing systems [16] created in 2019. Currently, on Linux systems with Intel chips (that support the RAPL power cap interface) and NVIDIA GPUs, it records: power draw from CPU and GPU, hardware information, python package versions, estimated carbon emission information, etc. To our knowledge, this is the most advanced tool which can be used by researchers to track their environmental impact.

From the four tools presented, only the *ML Emissions Calculator* and the Experiment Tracker can be used to monitor the carbon footprint of ML computations, they are summarised in Table 5.1.

Table 5.1: Existing tools to calculate the carbon footprint ML methods

	Experiment Impact Tracker	*ML Emissions Calculator*
Variables	Power consumed during run time, location with the IP address of the machine	Time, hardware type, cloud provider, location
Computational costs	Good performance thanks to the RAPL power monitoring interface	time-based tracker (see 3.1)
Communication costs	N/A	N/A
Precision of conversion into carbon footprint	Rough estimation except in California (California Independent System Operator's website) which displays real-time power grid conditions and greenhouse gas emissions)	Same rough estimation but provides raw and offsetted carbon emissions
Format	PyPI package	Webpage
High compatibility	Compatible with Linux systems running NVIDIA GPU's and Intel processors (which support RAPL)	Compatible with the provided list of hardware types, cloud providers and region of compute
Simple integration within ML projects	Can be installed using `pip` package-management system, see their documentation	All calculations need to be systematically performed manually and cannot be automated
Open to contribution	Yes (GitHub)	Yes (GitHub)
Release	2019	2019

The need for *CUMULATOR*

Both tools presented in Table 5.1 **lack the availability to compute the carbon footprint of communications** performed in the cloud, which we would like to implement. The *ML Emissions Calculator* was not designed to be integrated with other projects, but rather to provide an effortless way for every research group to roughly assess the environmental impact of their work. The *Experiment Impact Tracker* answers this integration need, but it requires a RAPL-compatible GPU, which in resource-limited settings may not the case. We would like to build a tool which can be RAPL or APM agnostic. Hence it could have been possible to extend their tools, but within this project, we decided to act fast, build and test our own one: *CUMULATOR*. It is presented in the next section.

5.2 Objective 2, Present *CUMULATOR* which will be used to assess the carbon footprint of ML methods in this project

We decided to build *CUMULATOR* to best answer the project requirements. To our knowledge, it is the second existing open-source tool available as a package to assess the environmental impact of ML methods.

5.2.1 Design choices

The design choices are summarised with respect to the requirements in Table 5.2.

Operational range and accessibility: the choice of the programming language. *CUMULATOR* could be coded in many different programming languages. We chose Python because it best suits the requirements of accessibility and operating range. Python can be defined as a high-level, interpreted, and general-purpose dynamic programming language that focuses on code readability. This makes it accessible even to a public with little programming knowledge such as students who will use it in their research at EPFL. Also, Python is by far the most popular programming language (31% of shares in 2020) and is on a rising trend according to the PYPL Index [34]. Hence, the probability that an ML framework is compatible with Python is maximised. It also benefits from extensive library support which enables the integration of *CUMULATOR* in many environments (*Experiment Impact Tracker*, *ML Emissions Calculator* are mainly coded in Python). Besides, it is interesting to add that Java was historically the most popular programming language (28% of shares in 2010), but its usage is decaying to the benefit of Python. The singularity point, where Python became more popular than Java, happened in mid-2018.

Looking at **high compatibility**, you need only the Python Standard Library to start doing

carbon footprint calculations.

Available for open usage: *CUMULATOR* is a package stored on PyPI and publicly available here.

Open to contributions: *CUMULATOR* source code is stored on GitHub inside the EPFL organisation called iGH (intelligent Global Health). Contributing to it is simple. As soon as you have the access rights, you can create your branch in the repository and work with your additional feature. The procedure to upload a new version online is the following:

1. Change version number in the setup file `setup.py`
2. Commit and push the changes to GitHub
3. Make a pull request on the master branch
4. Once the pull request is accepted, the host will push the changes to PyPI (or add you as a contributor to do it)

Table 5.2: Design choices for *CUMULATOR*

	CUMULATOR
Variables	Time (optional: reference values for hardware type, per-byte energy model, carbon intensity)
Computational costs	time-based tracker (see 3.1)
Communication costs	1byte energy model of *the Shift Project*
Precision of conversion into carbon footprint	Rough estimation
Format	PyPI package
High compatibility	Need only the Python Standard Library
Simple integration within ML projects	Can be installed using `pip` package-management system
Open to contribution	Yes (GitHub: iGH (intelligent Global Health))
Release	2020

5.2.2 List of assumptions

Preliminary remark: any change in those assumptions should be indicated by modifying the first digit of the version number. The last version used within this project is 0.0.5.

The goal of the project was to have an early-stage working version of *CUMULATOR* to integrate into *Alg-E* before the deployment of the platform in 2020-2021. As it relies on rough assumptions, the current version of *CUMULATOR* is best suited to benchmark ML methods. We do not aim to provide carbon footprint calculations with the state of the art accuracy. A couple of solutions presented in the last chapter would however rapidly improve this accuracy, and could be integrated later on.

Concerning the **cost of computations**, it is a time-based tracker. Hence, it assumes that 1) time is a good estimator of the energy efficiency and 2) that the TDP is a good estimator of the GPU's power consumption 3) the GPU is constantly under maximal theoretical load. To compute the energy consumption we take Nvidia GeForce GTX Titan X as GPU reference because it is used in the IC cluster of the EPFL Machine Learning and Optimization Laboratory (MLO). Its TDP is 250 W [12].

The **cost of communication** relies the "1byte model" of *the Shift Project* presented in section 3.2. From data showed on Figure 3.2 the most recent value is indeed, $6.894 \cdot 10^{-11}$ kWh/byte, and is from 2017. Hence we confirm that the "1byte model" (corresponding to $7 \cdot 10^{-11}$ kWh/byte, [24], "Materials") is based on data from 2017.

Following the material presented in 3.3, the average **carbon intensity** of electricity consumed at LV in Europe is chosen as reference value for the carbon intensity: 447 gCO_2eq/kWh [31].

5.2.3 How *CUMULATOR* works: concrete numbers

The following information is given according to the assumptions above of *CUMULATOR*. We shall precise that the calculation of the computation and communication costs is totally independent. One can perfectly use *CUMULATOR* to report one or the other type of cost only.

Concrete numbers:

- **One hour of GPU load is equivalent to 112 gCO_2eq**
- **1 GB of data traffic is equivalent to 31 gCO_2eq**
- One hour of GPU load is equivalent in emissions to communicate a file of 3.6 GB using data center network

Calculation of the computation costs

As explained hereafter, each instance of the *CUMUALTOR* class automatically records and agglomerates the time whenever the function `off()` is used. The equation below shows how the

carbon footprint of computations is calculated.

$$Computation\_costs = Time(s) \cdot \frac{GPUload(0.250kW) \cdot Carbon\_intensity(447g\mathrm{CO}_2eq/kWh)}{3600s/h} \tag{5.1}$$

Calculation of the communication costs

Similarly, each instance of the *CUMUALTOR* class automatically records and agglomerates the file sizes whenever the function `data_transferred(file_size)` is used. The equation below shows how the carbon footprint of communications is calculated.

$$\begin{aligned} Communication\_costs = File\_size(bytes) \cdot 1byte\_energy\_model(6.894 \cdot 10^{-11}kWh/byte) \\ \cdot Carbon\_intensity(447g\mathrm{CO}_2eq/kWh) \end{aligned} \tag{5.2}$$

5.2.4 How to use *CUMULATOR*

Instead of making a documentation, we commented the source code and present how to use *CUMULATOR* in this section. The entire source code is embedded in `src/base.py`. *CUMULATOR* consists of one class in which a set of members and methods activate, deactivate a chronometer, record time and size of files communicated, and finally compute and display the relevant carbon footprints based on the assumptions (see 5.2.2). Computation and communication costs are independently derived to let the user look at only one aspect if the other is not applicable. The most important features of *CUMULATOR* are explained below:

Install it using `pip install cumulator`. Import the script with `from cumulator import base`. Create class a instance with `cumulator = base.Cumulator()`. If you run your work in parallel on `n` GPUs, the carbon footprint should be multiplied by the number of GPUs. You can modify this number with `cumulator.n_processors = <number of GPUs>`.

Cost of computations. Activate or deactivate chronometer by using `cumulator.on()`, `cumulator.off()` whenever you perform ML computations (typically within each interation). It will automatically record each time duration in `cumulator.time_list` and accumulate it in `cumulator.cumulated_time`. Then return carbon footprint due to all computations using `cumulator.computation_costs()`.

Cost of communications. Each time your models sends a data file to another node of the network, record the size of the file which is communicated using `cumulator.data_transferred(file_size)`. The amount of data transferred is automatically recorded in `cumulator.file_size_list` and accumulated in `cumulator.cumulated_data_traffic`. Then return carbon footprint due to all communications using `cumulator.communication_costs()`

Return the **total carbon footprint** using `cumulator.total_carbon_footprint()`. You can also display the carbon footprint in terminal using `display_carbon_footprint()`

5.3 Objective 3, Integrate *CUMULATOR* to the *Alg-E* platform

Alge-E is the first platform where *CUMULATOR* was integrated. Since all computations are made locally, *Alg-E* reports the carbon footprint linked to computations. It does not perform any communication. Hence, the validation of *CUMULATOR* concerns only the computational costs. In this section, we present our analysis, describe the results, and then present the extensions added to *Alg-E*.

5.3.1 Compute the carbon footprint of each model under various conditions

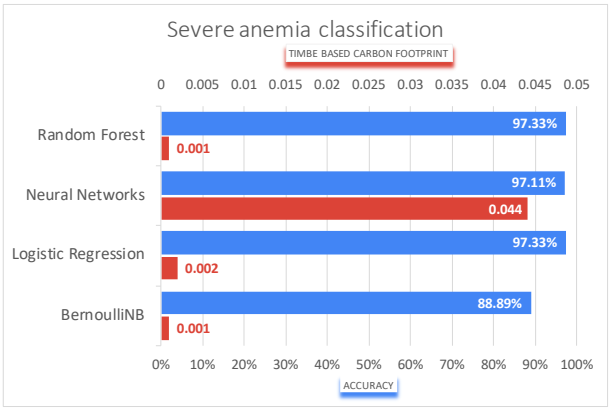
All the data present below was extracted from the *Alg-E* platform after training the models. Each time we compare the different models to investigate the objectives presented in 4.3.2. We plot the time-based carbon footprint (in gCO_2) and the accuracy (in percentage) on a different axis. It is important to note that the scale of the carbon footprint varies from one graph to another, this is why we display the label values directly in or next to the bars. We wrote self-explicit captions to ease the understanding during the analysis in the next section.

Results of classifications

The results are presented in Figures 5.1, 5.2 and 5.3. To examine the accuracy and the time-based carbon footprint trade-offs, the Table 5.3 shows the averaged values of those metrics for the three classification tasks.

Table 5.3: Average values of accuracy and time based carbon footprint per model for the three classification tasks (without oversampling where applicable)

Model	**Accuracy**	**Time-based carbon footprint in gCO_2eq**
Random Forest	97.76%	0.21
Neural Network	97.71%	2.9
Logistic Regression	93.06%	0.83
BernoulliNB	71.63%	0.14



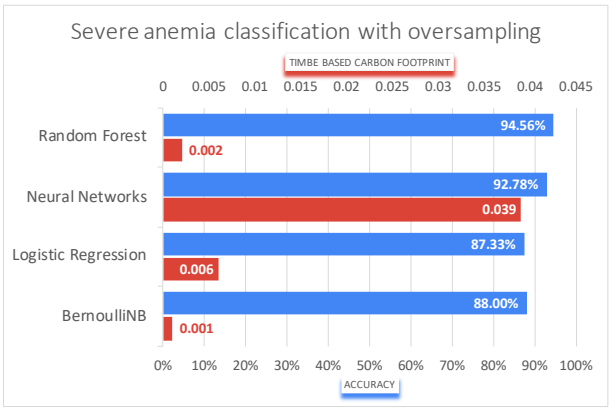
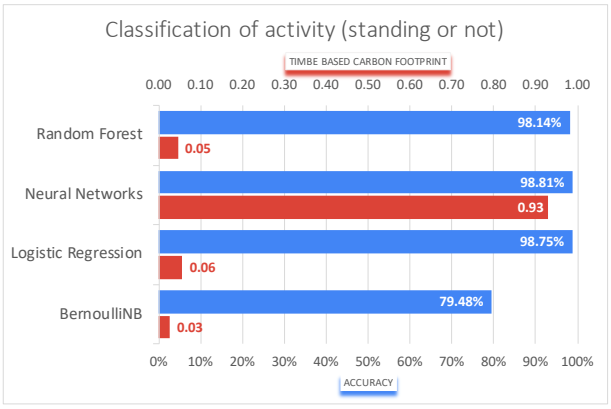


Figure 5.1: Severe anemia binary classification diagnosis on the ePOCT dataset, with and without oversampling (3'192 patients and 41 features)



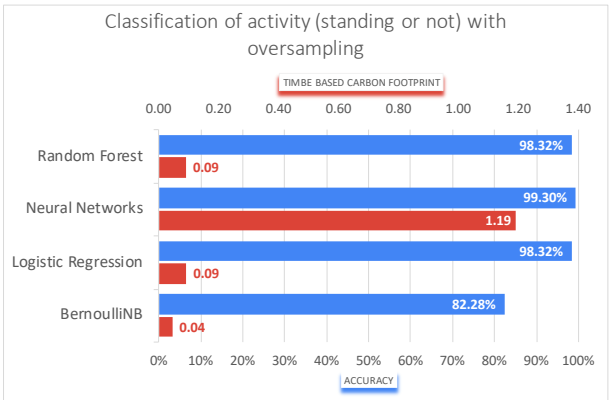


Figure 5.2: Binary classification of standing or non-standing activity on the Standing Posture dataset, with and without oversampling (10'929 activities and 561 features)

Results of regressions

We investigated the evolution of the carbon footprint with the number of features for the dataset with CT images (Figure 5.5). The regression was performed to predict the axial location of the image based on the histograms values of bone structure and air inclusions.

5.3.2 Analysis of *Alg-E* results

One should observe the plots while keeping in mind that the ideal model would have the highest accuracy (or lowest MSE) while minimising the carbon footprint. We derive general conclusions from the observation of the results and to answer the objectives of the analysis presented in 4.3.2.

Cost differences between the classification models:

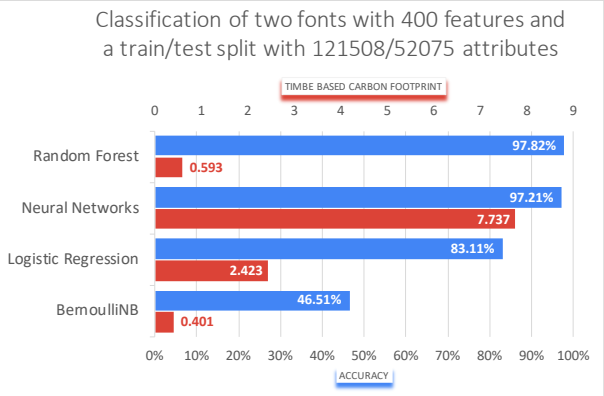


Figure 5.3: Classification of fonts (OCR-B or Segoe) using pixel data of characters (173'583 images and 400 features)

- Neural Network seems to be systematically the most costly model in terms of carbon footprint (Table 5.3)
- The implementation of the Naive Bayes model seems to have poor performances (Table 5.3. It constantly has the worse accuracy, ranging from 46% to 89% (Figures 5.1, 5.2, 5.3)
- Random Forest and Neural Network seem to be the most accurate models with an average accuracy over 97% (Table 5.3)
- Random Forest clearly has the best accuracy vs carbon footprint trade-off. It is in average 10 times less costly than Neural Network with similar accuracy (Table 5.3)

Cost differences between regression models looking at Figure 5.4:

- Random Forest largely outperforms all the model implementations (the MSE is minimum 20 times lower) but at the cost of high carbon footprint (between 10 and 20 times higher costs)
- Linear Regression model does not perform well as we obtain MSE of the order of 10^{21} superior to the values of the other models. This is most probably because the data is nonlinear whereas the model can only perform a linear fit.

Benefit of feature selection methods. Figure 5.4 is a good example of the drawback of using feature selection. In this case, the feature selection provides worsened results. We decided to investigate the evolution of the carbon footprint with the number of features for this regression case using the best matching model (Random Forest). We ran the same dataset again twice by deleting one feature every two and then one every three. An even better approach would have been to randomly select a portion of the features many times and average the results. Figure 5.5 shows a linear fit, meaning that each additional feature adds approximately 0.0045 of gCO_2eq during training.

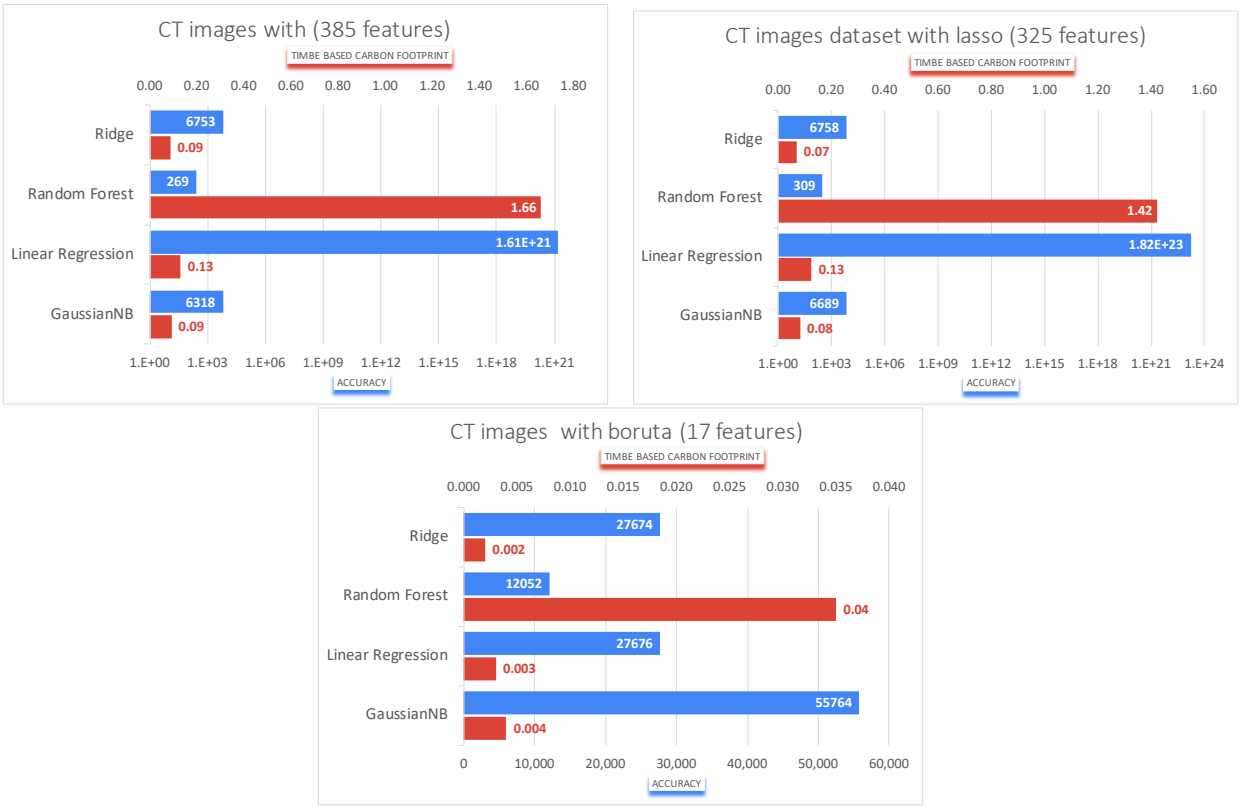


Figure 5.4: Prediction of the location of the image, based on histograms of the bone structure and air inclusions of human patients

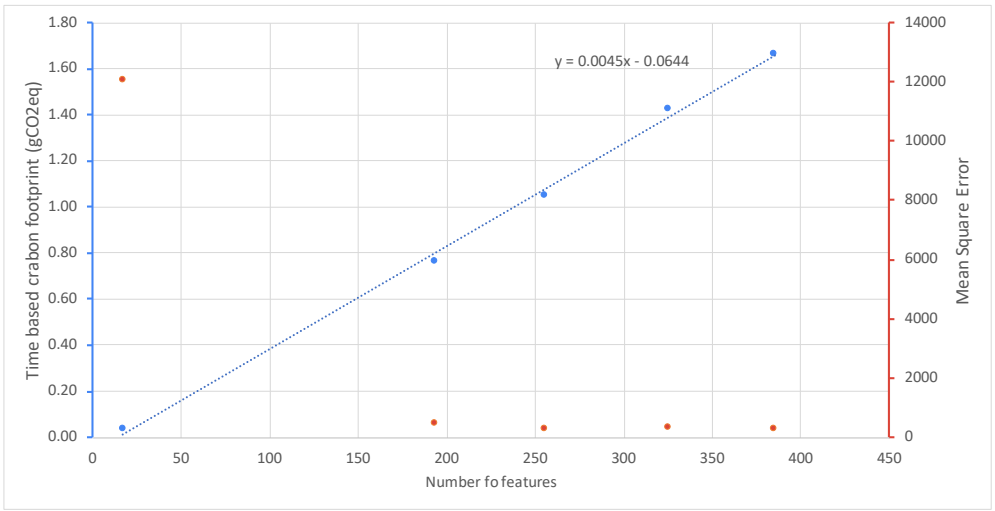


Figure 5.5: Evolution of the time-based carbon footprint and accuracy with the number of features for a regression task with 53'500 data points using Random Forest

The **benefit of oversampling** is not clear. Depending on the models on Figures5.1 and 5.2, it leads to an increase and decrease of the accuracy and carbon footprint. More investigations

would be necessary to provide clear statements.

General conclusions:

- We provide relevant examples of accuracy vs carbon footprint trade-offs. Figure 5.4 shows that Random Forest is the best model for the regression task on the CT images dataset. However, looking at Table 5.3, the model which has the highest accuracy shouldn't systematically be chosen. Random Forest and Neural Network have close accuracy but divergent carbon footprints
- The *Alg-E* platform seem to be polyvalent and able to fit linear and non-linear datasets with high dimensionality and high number of features (up to 170k data points and 500 attributes)

5.3.3 *Alg-E* extensions: communicate the trade-off between accuracy and carbon footprint on the user interface

Models for predict dem_age_int

model	started_at	MSE	MAE	Carbon footprint
GaussianNB (Regression)	June 17, 2020, 9:11 p.m.	41815.031	1709.395	0.101
Linear Regression (Regression)	June 17, 2020, 9:11 p.m.	2251.859	374.946	0.128
Random Forest (Regression)	June 17, 2020, 9:11 p.m.	2440.269	320.327	0.245
Ridge (Regression)	June 17, 2020, 9:11 p.m.	2238.002	373.785	0.040

General Information

No feature selection method was applied and the models ran with 44 features.
No filter was applied to the dataset and the records were used to train the model and 958 were used to validate the models.

Models for predict dxfinal_anemia_sev

model	started_at	AUC	Accuracy	Carbon footprint	Recall	Precision	F1Score
BernoulliNB (Classification)	June 17, 2020, 11:32 p.m.	48.737%	96.66%	0.043	0.0%	0.0%	0.0%
Logistic Regression (Classification)	June 17, 2020, 11:32 p.m.	68.645%	99.269%	0.105	37.5%	60.0%	46.154%
Neural Networks (Classification)	June 17, 2020, 11:32 p.m.	49.895%	98.956%	0.311	0.0%	0.0%	0.0%
Random Forest (Classification)	June 17, 2020, 11:32 p.m.	56.25%	99.269%	0.060	12.5%	100.0%	22.222%

General Information

No feature selection method was applied and the models ran with 44 features.
No filter was applied to the dataset and the records were used to train the model and 958 were used to validate the models. The split is stratified by the response variable.

Figure 5.6: Example of the updated table within the `Models` menu of *Alg-E* with the (time based) carbon footprint for a regression (up) and a classification (below) task

Upon the integration of *CUMULATOR*, we did two main changes on the *Alg-E* user-interface:

1. We decided to display the carbon footprint in the `Models` and `Runs` tabs in the menu. The Figure 5.6 shows how the updated table looks like for regression and classification tasks and the procedure to plot it.

2. We improved the visualisation of the accuracy/MSE and carbon footprint trade-off by displaying a bar chart in a new menu called `Benchmark` as shown in Figure 5.7. We defined the best matching unit as the model which has the highest accuracy (or lowest MSE) between all runs. We plot this performance metric along with the corresponding carbon footprint in a horizontal bar chart and we give 1) the set of hyperparameters related to this best matching unit, and 2) a hyperlink redirecting to the model for deeper investigation. By the way, this extension also lets the user know what is the set of hyperparameters (oversampling, feature selection) which led to the best accuracy/MSE. Before this extension, there was no simple way to know which model was the best while looking at a specific dataset and target label.

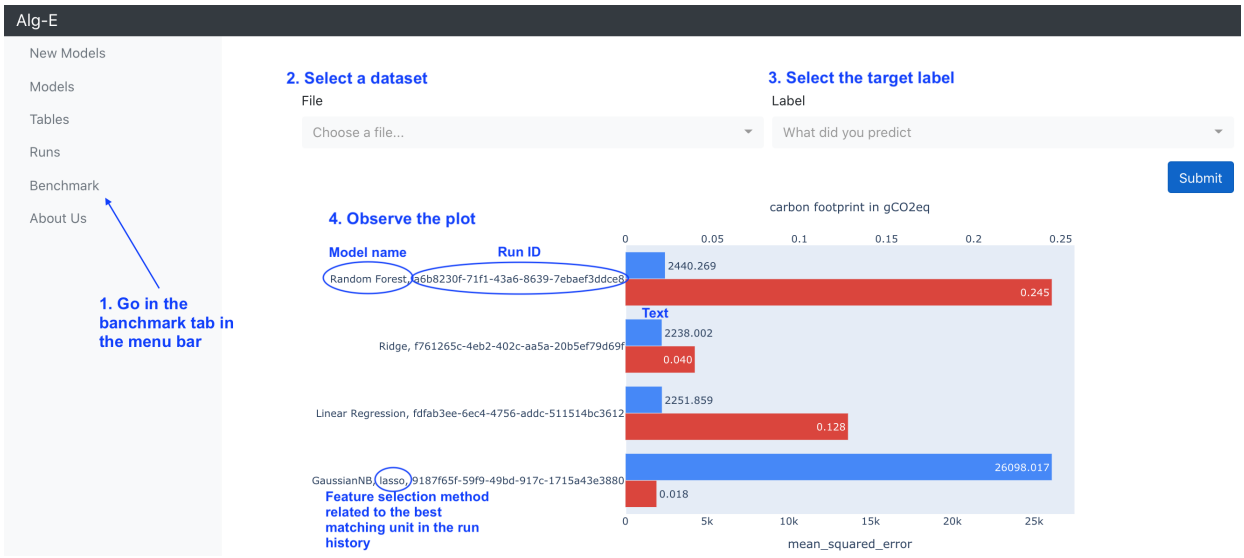


Figure 5.7: Chart displaying the highest accuracy within all runs of each model and the related carbon footprint and the steps to plot it

5.4 Objective 4, Generate a list of features required to calculate the carbon footprint of a large-scale medical data analysis platform, such as will be used in the *DYNAMIC* trial

We have extracted five generic actions from the analysis of *DYNAMIC*'s workflow in Figure 4.3. Table 5.4 lists the features required to calculate the carbon footprint for each generic action. The column "Ref." links each activity to a number in the worklow diagram.

Table 5.4: List of the features required to calculate the carbon footprint of *DYNAMIC* during operations

Type of activity	**Ref. (Fig. 4.3)**	**Features to report**	**Method for carbon footprint assessment**
Healthcare	1	1) Time and location of the consult 2) Number of consults 3) Hardware type (smartphone, laptop, or desktop computer)	Screening time is calculated
Communication	2, 5, 6, 7	1) Amount of data 2) Type of network (Wired network, WIFI network, Mobile network, Data center network) 3) Location of emitting and receiving point 4) Name of the cloud provider (if applicable)	Using the per-byte energy model from *the Shift Project* (see [24], "Materials", "1byte model")
ML use (*Alg-E*)	1	Computations and communication during training of ML models	Using *CUMULATOR*
ML research	8	Computations and communication during training of ML models	Using *CUMULATOR*
Personal work	all	Data traffic when using your browser	Carbon Impact Statement (see protocol in 5.5.2)

5.5 Objective 5, Create a protocol to help future researchers and students quantify the carbon footprint of their projects

A protocol for systematic reporting of carbon footprints in scientific research has been set up by the *Experiment Impact Tracker* team ([16]). This script accumulates the carbon footprint related to the values recorded with the *Experiment Impact Tracker*. We suggest creating a protocol that incorporates carbon emissions of projects as a whole. It shall not only track the cost of the ML computations but also the cost of web browsing activity, virtual meetings, screening time, etc. This will allow us to raise awareness of the environmental costs of day-to-day activities. The

protocol could also be extended to non-ML-related projects later on.

5.5.1 Variables influencing the carbon footprint of a research project

The carbon footprint of your research project mainly depends on: your mean of communication (virtual meeting), your activities on your computer (screening time, online searches, size of downloads), your ML simulations (GPU cluster load). We justify the numbers which we will use:

- **ML simulations**: the carbon footprint can be estimated with *CUMULATOR*, see 5.2.4.
- **Data downloaded while browsing**: use the *Carbonalyser* add-on presented in 5.1.
- **Screening time**: *Carbonalyser* records the number of minutes of browsing. We assume that the power consumption of a laptop is 40W (as a comparison, the one of a desktop PC is 150W) [32] to derive the energy consumption.
- **Virtual meetings**: a study from 2014 analysed the life cycle of network and video-conferencing terminal equipment [32]. The total power consumption is 187.6 W for the following equipment: laptop, entry-level CODEC, camera, sound system, microphone, home/office LAN. We take a default value of 45 minutes per weekly meeting.
- The energy to carbon footprint conversions will be done with the average carbon intensity of electricity consumed at LV in Europe: 447 gCO_2eq/kWh [31] (see 3.3 for explanation).

To provide tangible information, this scripts also computes the number of kilometers a car would have run to produce an equivalent amount of greenhouse gas emissions. We base this calculation on the average carbon dioxide emissions from new passenger cars registered in the European Union in 2018: 148.1 gCO_2eq/km [30]. A report from 2011 suggests that such values are indeed a measure of the carbon footprint equivalent for the greenhouse gas emissions of a typical car (and do not represent only carbon dioxide emissions) [36].

5.5.2 The Carbon Impact Statement Protocol

For this protocol, you will need to use a dedicated browser only for the activities related to your project. We built `bonus.py`, a script inside *CUMULATOR*, to easily compute the carbon footprint of your research in one-shot at the end of your project. Here's the protocol to we came up with:

1. At the beginning of your project:
   (a) Prepare your project web browser: 1) select either Chrome or Firefox dedicated to the project, 2) make sure you clear your history, and 3) install *Carbonalyser*.

(b) Install *CUMULATOR* (explained in 5.2.4). Note that at no point will any of your search history be communicated to the lab. This project respects privacy and all reporting is entirely user-controlled.

2. During the project: use *CUMULATOR* to record the carbon footprint of ML models (if applicable)

3. At project end:

   (a) Recover data from your web browser by running *Carbonalyser* (can take several minutes)

   (b) Estimate your weekly screening time. *Carbonalyser* records the time when activated, but that time does not impact the carbon footprint. In other words, it only looks at data traffic, not at the power drained to run your laptop.

   (c) Import the script with `from cumulator import bonus`

   (d) Run `bonus.project_carbon_footprint(carbonalyser, ml_simulations, screening_time, n_weeks=14, meeting_duration=45, hardware_consumption=40)` with the required function arguments. Put zero where not applicable. Explanation of the function argument:

   - `carbonalyser` is the data from 3.(a)
   - `ml_simulations` is the total carbon impact recorded using *CUMULATOR*
   - `screening_time` is the estimated work time on your laptop (or other hardware) per week in seconds
   - `n_weeks` is the number of weeks of the project
   - `meeting_duration` is the average duration of a weekly meeting
   - `hardware_consumption` is the power consumption of your laptop (or other hardware you work on) in watts

   (e) Screenshot the output and insert it in your report

5.5.3 Carbon Impact Statement of this project

This project was used as an experiment to create the protocol. We tested different tools and used the presented protocol since the fourth week after the beginning of the project. This work consumed the equivalent of 7.31 kg of CO_2 emissions equivalent to driving a car over 49 km, see Figure 5.8.

Looking at our results, the internet data traffic was the biggest source of carbon emissions equivalent. This is certainly because of the low amount of time used to run ML models (the *Alg-E* platform was designed to run all models under 5 seconds). From 5.2.3, the internet data traffic consumed during this project would be equivalent to roughly 60 hours of GPU load.

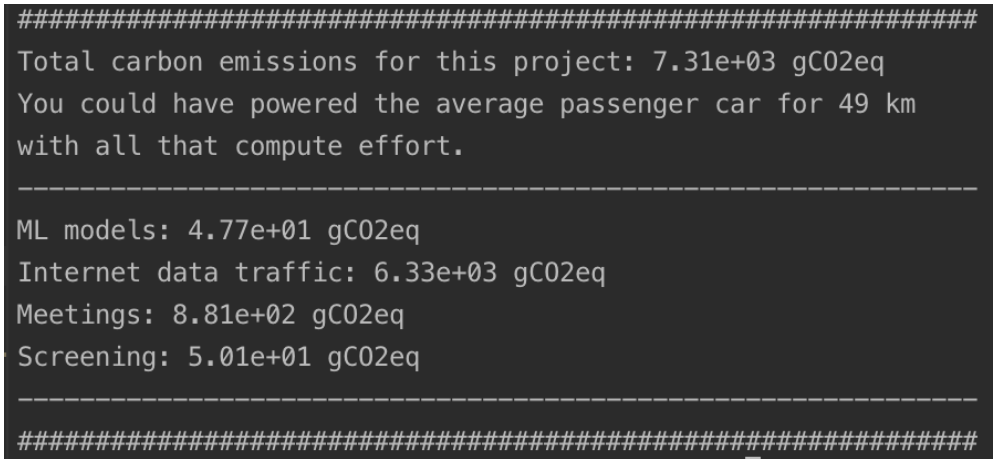


Figure 5.8: Carbon footprint of this project

According to Martin students usually accumulate 100 hours of GPU load during their ML semester projects at the MLO, which is equivalent to 11 kilograms of carbon dioxide emissions. As a conclusion, in terms of carbon footprint, the internet data traffic during a semester project usually scales roughly with one-half of the ML simulations.

5.5.4 Tips to become an environmentally conscious digital citizen

The principle of digital sobriety was elaborated by *the Shift Project* to raise awareness about the possibility to not use energy-intensive digital technologies where possible and to use your devices with parsimony. Many practices can be found online, but most of them are very restrictive (e.g. smaller screens consume much less power). Let's start simple, here are two practices to become an environmentally conscious digital citizen:

1. **Eliminate vampire power**: when electronic devices are put in stand by mode or even switched off, some electricity is drained from the grid (called vampire power) only because they are connected to it.
2. **Choose a sustainable web browser**. Ecosia is an exemplar of this model. Their philosophy is "we plant thanks to your searches". They planted 50 million trees in five years. In February 2020 they made 2.5 million and invested 71% of their incomes into tree planting or other projects such as electricity production from solar power. They make money through ads which appear above and below the research results and by selling products (1 T-shirt plants 20 trees!). Their success led them to diversification. They now fight climate change by removing CO2 from the atmosphere and accelerate the energy transition away from fossil fuels by adding solar energy to the electricity grid.
3. Use ***Carbonalyser*** presented in section 5.1.

5.6 The logo: an allegory of this project's aim

Images are the best vector to raise awareness. Hence, we decided to design a logo for *CUMULATOR* that personifies the aim of this project (see Figure 5.9). The origin of the environmental impact of machine learning applications is the GPU load needed to train the model. This is why we choose a drawing of a processor as the main icon. Now, the electricity used to run those GPUs is produced with a certain carbon footprint, which equivalent to releasing pollutant greenhouse gases in the atmosphere, hereby represented by the smoke. The name of *CUMULATOR* comes in: this API can assess the carbon footprint of ML computations and communication. We can, therefore: 1) raise awareness about the carbon footprint of digital technologies, and 2) optimise ML methods to mitigate the amount of greenhouse gases generated. Finally, the earth, the plant, and the green atmosphere denotes the commitment of this work for sustainable-AI, which is a digital key-word for fighting climate change.

Figure 5.9: Logo of *CUMULATOR*

Discussion

The work in this project was multi-disciplinary. We explained and presented the key material needed to calculate the carbon footprint of ML methods. After analysing the existing tools to calculate the carbon footprint, we decided to create *CUMULATOR* as the best fit for the project requirements. The benefit of integrating this new API into *Alg-E* is that it forced us to go rapidly from design to implementation, and open-source deployment, and it ensured to produce a working API.

Concerning academia, we decided to use this project as an experiment to build a protocol to systematise the report of carbon footprint. We implemented the Carbon Impact Statement Protocol as a "bonus" script within *CUMULATOR*. It hereby benefited from the fact that *CUMULATOR* had to be open source for more important reasons linked to the *DYNAMIC* trial. We should point out that at the beginning of the project we were not sure about the feasibility of this protocol. It is now a scalable tool that can simply be installed and used by every student and researcher at EPFL and even in the world.

The main disadvantage of the outcome of the project is due to the high amount of different tasks performed. A lot of work can be done to improve *CUMULATOR* and to study in-depth the efficiency of the *Alg-E* platform. Eventually listing the features to calculate the carbon footprint of the *DYNAMIC* trial is a first step that will be useful only if further work is performed. We will discuss this in the next and final chapter.

Limitations and future work

The continuity of the project has two clear aspects. Firstly, the accuracy of the carbon footprint estimations can be improved and the interface can be adapted upon request of future users. Secondly, *CUMULATOR* and the Carbon Impact Statement Protocol where created to be deployed in-the-field. The purpose of *CUMULATOR* is to fill the gap between the acknowledgment of greenhouse gas emissions and the dedication to take the environmental impact into account while designing the next generation of ML systems. The Carbon Impact Statement Protocol was designed to make students and researchers realise the carbon footprint of their work, thus bringing climate change considerations into individual academic projects.

7.1 Next improvements for *CUMULATOR*

The main limitation of *CUMULATOR* lies in the fact that any evaluation of the carbon footprint is a rough estimation. This is due in particular to the lack of data about carbon intensity (in gCO_2eq/kWh), which is a critical step in the determination of the carbon footprint (see 3). The electricity map (selected by the two tools in 5.1) provides carbon intensity estimations at a country or regional scale and in less than 50 countries in the world (none in Africa). Observing that the dispersion of the carbon intensity values in the world ranges from 13 to 820 gCO_2 eq, makes this data even more crucial (Figure 3.3). The absolute value of any carbon footprint estimation in ML is biased. Hence, the best comparisons of the environmental impact of ML methods are done within the same geographic location.

However to derive estimates of the carbon footprint with maximal precision, *CUMULATOR* can be extended with 5 action items listed below. Generally, we should follow the evolution of the research in the carbon footprint of ML.

Action 1: integrate the *Experiment Impact Tracker*.
Expected impact: provide better energy consumption estimates at run time whenever the hardware is compatible thanks to the RAPL interface.

Action 2: add a database of the energy consumption of commonly used processors to improve the accuracy of the time-based calculation of carbon footprint. The study from *ML Emissions Calculator* can be used as a model. [23]

Expected impact: improve the time-based energy tracking

Action 3: add the geographical distribution of carbon intensity of electricity drained 1) from grids and 2) within data centers for each cloud providers 3.3, [23]
Expected impact: improve the precision of the energy to carbon footprint conversions.

Action 4: regularly update all heuristics of *CUMULATOR* to stay as realistic as possible. For example, since data centers are becoming energy efficient with time, the 1byte energy model could be halved between 2017 and 2025 see Figure 3.2, the values used to compute the communication cost of ML should be periodically updated. *the Shift Project* indeed announced in March 2020 that their analysis is going to be updated within a couple of months.
Expected impact: maintain realistic carbon footprint estimations.

Action 5: closely follow up the future publications about methods to calculate and tools to efficiently communicate the carbon footprint in ML. The *Experiment Impact Tracker* is part of a "working paper" published on 14th February 2020 [16]. This suggests that it is a hot topic subject to many contributions in the upcoming years.
Expected impact: make the right choices for *CUMULATOR* and let it be updatable.

7.2 Towards environmentally conscious research projects at MLO or EPFL

This project gave birth to a new tool to compute the carbon footprint of ML methods and a protocol to report it in academia. We identified the 3 next steps to empower those fresh creations:

Use-case 1 of *CUMULATOR*: optimise the efficiency of communications in various multi-node ML models.

Use-case 2 of *CUMULATOR*: add the carbon footprint in the list of the performance parameters on MLBench projects. At the moment the computational costs are time-based. Since time is already recorded in the framework, the carbon impact of computations will be proportional to it. However, observing the difference of carbon footprint between computation and communications is a new interesting information available with *CUMULATOR*

Deploy the Carbon Impact Statement Protocol: to assess the carbon footprint of research projects within EPFL. We suggest to start with projects from the MLO to confirm if other students are satisfied with it. The protocol could be extended later on to all projects at EPFL.

7.3 Next steps to systematise the carbon footprint recordings and reporting during the *DYNAMIC* trial

Here is a proposition for the next steps of the *DYNAMIC* team to manage the environmental impact of the trial:

1. Look at the results of Objective 4 (see 5.4)
2. Investigate the feasibility to report each activity's features
3. Discuss the adaptations or solutions that can be provided, and update the Table 5.4.
4. Select the activities that we wish to implement and integrate them into the trial

Update on 17.06.2020: we presented (Annie and Tristan) the results to the IT manager of *DYNAMIC* (Alan). Recording the carbon footprint seems to be feasible for the following activities: Healthcare, Communication, ML use (*Alge-E*), and ML research. We agreed that their task will be to accumulate the data (except for ML research) and send it to us (the iGH team in the MLO). Our responsibility will be to 1) add the carbon footprint linked to ML research (sum the Carbon Statements of the related projects including this one) and 2) compute the total carbon footprint of the *DYNAMIC* trial.

This will be implemented as a "before and after" study to assess the impact of communicating the carbon footprint to researchers. Here, carbon footprint will be calculated but not shared in the first half of the study and then calculated and shared live in the second half. Structured questionaires of knowledge, attitudes and practices (KAP) will be performed on researchers in both sessions.

7.4 Limitations of *Alg-E*

The main limitations of our analysis in section 5.3, is that we only looked at accuracy/MSE and carbon footprint. In reality, depending on the objective of the medical analysis, observing the trade-offs between all the performance parameters including the ROC curve, precision, recall, and F1 score is crucial.

We reported several bugs and remarks on *Alg-E*:

- In the `New Models` tab, if someone changes their mind and de-selects a label (label1) to

re-selects another label (label2), the menus `Models` and `Metric` provides the choices linked to label1 and label2. Hence, one can run a regression model on a classification dataset.

- My local implementation of *Alg-E* was shutting down after several minutes of computations when I tried to upload tables over 300 MB.
- The report [1] states that all models should run under 5 seconds. During my tests, Neural Network ran for 38 seconds.
- Sometimes the differentiation between categorical and numerical data is blurry. If an attribute has 2 different integers, it will be considered as categorical data, however, an attribute with 100 different integers is considered as numerical data. Where's the limit?

Thank you for reading this report!
May you have any feedback or remarks, I am pleased to discuss with you (tristan.trebaol@epfl.ch).

Bibliography

[1] "A machine learning platform to generate interpretable medically approved, data-responsive clinical Decision algorithms". In: *Master Thesis - EPFL* (2020).

[2] *AMD vs Intel Market Share*. URL: https://www.cpubenchmark.net/market_share.html.

[3] Anders Andrae and Tomas Edler. "On Global Electricity Usage of Communication Technology: Trends to 2030". In: *Challenges* 6.1 (2015), pp. 117–157. DOI: 10.3390/challe6010117.

[4] *Artificial Intelligence Index Report 2019*. URL: https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf.

[5] Kristen W. Carlson. "Safe Artificial General Intelligence via Distributed Ledger Technology". In: *Big Data and Cognitive Computing* 3.3 (Aug. 2019), p. 40. DOI: 10.3390/bdcc3030040.

[6] Dario Caro. "Carbon Footprint". In: *Encyclopedia of Ecology* 4 (2019), pp. 252–257. DOI: 10.1016/b978-0-12-409548-9.10752-3.

[7] Intergovernmental Panel On Climate Change. "Climate Change 2014 Mitigation of Climate Change". In: (2014), pp. 511–589. DOI: 10.1017/cbo9781107415416.

[8] *Cisco Global Cloud Index (GCI) 2014–2019*. URL: http://www.audentia-gestion.fr/cisco/Cloud_Index_White_Paper.pdf.

[9] *Climate Change Evidence: How Do We Know?* Dec. 2019. URL: https://climate.nasa.gov/evidence/.

[10] Bogdan Florin Cornea, Anne-Cecile Orgerie, and Laurent Lefevre. "Studying the energy consumption of data transfers in Clouds: the Ecofen approach". In: *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)* (2014). DOI: 10.1109/cloudnet.2014.6968983.

[11] *Fascinating Google Search Statistics (Updated 2019)*. Mar. 2020. URL: https://seotribunal.com/blog/google-stats-and-facts/.

[12] *GeForce GTX TITAN X Graphics Card*. URL: https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications.

[13] *Global Greenhouse Gas Emissions Data*. Sept. 2019. URL: https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data#Sector.

[14] Daniel Hackenberg, Thomas Ilsche, Robert Schone, Daniel Molka, Maik Schmidt, and Wolfgang E. Nagel. "Power measurement techniques on standard compute nodes: A quantitative comparison". In: *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (2013). DOI: 10.1109/ispass.2013.6557170.

[15] Daniel Hackenberg, Robert Schone, Thomas Ilsche, Daniel Molka, Joseph Schuchart, and Robin Geyer. "An Energy Efficiency Feature Survey of the Intel Haswell Processor". In: *2015 IEEE International Parallel and Distributed Processing Symposium Workshop* (2015). DOI: 10.1109/ipdpsw.2015.70.

[16] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. *Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning*. 2020. arXiv: 2002.05651 [cs.CY].

[17] Urs Hölzle. *Powering a Google search*. Jan. 2009. URL: https://googleblog.blogspot.com/2009/01/powering-google-search.html.

[18] *How much carbon dioxide is produced per kilowatthour of U.S. electricity generation?)* URL: https://www.eia.gov/tools/faqs/faq.php?id=74&t=11#:~:text=In%202018,%20total%20U.S.%20electricity,of%20CO2%20emissions%20per%20kWh..

[19] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R. Ganger, and Phillip B. Gibbons. "Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds". In: *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI '17)* (Mar. 2017).

[20] Kai Hwang. *Cloud computing for machine learning and cognitive applications*. The MIT Press, 2017.

[21] Kristina Keitel, Frank Kagoro, Josephine Samaka, John Masimba, Zamzam Said, Hosiana Temba, Tarsis Mlaganile, Willy Sangu, Clotilde Rambaud-Althaus, Alain Gervaix, and et al. "A novel electronic algorithm using host biomarker point-of-care tests for the management of febrile illnesses in Tanzanian children (e-POCT): A randomized, controlled non-inferiority trial". In: *PLOS Medicine* 14.10 (2017). DOI: 10.1371/journal.pmed.1002411.

[22] Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen, and Zhonghong Ou. "RAPL in Action". In: *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 3.2 (2018), pp. 1–26. DOI: 10.1145/3177754.

[23] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. "Quantifying the Carbon Emissions of Machine Learning". In: *arXiv preprint arXiv:1910.09700* (2019).

[24] *Lean ICT, TOWARDS DIGITAL SOBRIETY*. Mar. 2019. URL: https://theshiftproject.org/en/lean-ict-2/.

[25] XIAOLIN LI. *Cloud computing for data-intensive applications*. SPRINGER-VERLAG NEW YORK, 2016.

[26] Carbon Footprint Ltd. *Carbon Calculator*. URL: https://www.carbonfootprint.com/calculator.aspx.

[27] F. Mboussou, P. Ndumbi, R. Ngom, Z. Kassamali, O. Ogundiran, J. Van Beek, G. Williams, C. Okot, E. L. Hamblion, B. Impouma, and et al. "Infectious disease outbreaks in the African region: overview of events reported to the World Health Organization in 2018". In: *Epidemiology and Infection* 147 (2019). DOI: 10.1017/s0950268819001912.

[28] "Methodological Aspects of Carbon Footprint". In: *The Carbon Footprint Handbook* (2015), pp. 1–2. DOI: 10.1201/b18929-2.

[29] *MLBench*. URL: https://mlbench.github.io/.

[30] *Monitoring of CO2 emissions from passenger cars – Regulation (EU) 2019/631*. June 2020. URL: http://co2cars.apps.eea.europa.eu/.

[31] Alberto Moro and Laura Lonza. "Electricity carbon intensity in European Member States: Impacts on GHG emissions of electric vehicles". In: *Transportation Research Part D: Transport and Environment* 64 (2018), pp. 5–14. DOI: 10.1016/j.trd.2017.07.012.

[32] Dennis Ong, Tim Moors, and Vijay Sivaraman. "Comparison of the energy, carbon and time costs of videoconferencing and in-person meetings". In: *Computer Communications* 50 (2014), pp. 86–94. DOI: 10.1016/j.comcom.2014.02.009.

[33] Juan Manuel Paniego, Silvana Gallo, Martín Pi Puig, Franco Chichizola, Laura De Giusti, and Javier Balladini. "Analysis of RAPL Energy Prediction Accuracy in a Matrix Multiplication Application on Shared Memory". In: *Communications in Computer and Information Science Computer Science – CACIC 2017* (2018), pp. 37–46. DOI: 10.1007/978-3-319-75214-3_4.

[34] *PYPL PopularitY of Programming Language index*. URL: http://pypl.github.io/PYPL.html.

[35] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. "Green AI". In: (July 2019).

[36] Brian Shorter. *Guidelines on greenhouse gas emissions for various transport types*. Mar. 2011.

[37] Emma Strubell, Ananya Ganesh, and Andrew Mccallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019). DOI: 10.18653/v1/p19-1355.

[38] "The cost of training machines is becoming a problem". In: *The Economist* (June 2020). URL: https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem.

[39] *Thermal Design Power (TDP) in Intel® Processors*. URL: https://www.intel.com/content/www/us/en/support/articles/000055611/processors.html.

[40] *Three African countries halt polio outbreaks*. URL: https://www.afro.who.int/news/three-african-countries-halt-polio-outbreaks.

[41] *Total greenhouse gas emissions (kt of CO2 equivalent)*. URL: https://data.worldbank.org/indicator/EN.ATM.GHGT.KT.CE.

[42] *U.S. Energy Information Administration - EIA - Independent Statistics and Analysis*. URL: https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a.

[43] *Understand and optimize the carbon footprint of electricity*. URL: https://api.electricitymap.org/?utm_source=electricitymap.org&utm_medium=referral.