A Methodology and a Toolbox to Explore Dataset related to the Environmental Impact of HTTP Requests

Christophe Cérin, Mathilde Jay, Laurent Lefèvre

A Methodology and a Toolbox to Explore Dataset related to the Environmental Impact of HTTP Requests

Christophe Cérin
LIPN, UMR CNRS 7030
University Sorbonne Paris-Nord
99, avenue J.B. Clément
93430 Villetaneuse, France
christophe.cerin@univ-paris13.fr

Mathilde Jay
LIG
University Grenoble Alpes
150, avenue du Torrent
38401 Saint Martin d'Hères
mathilde.jay@inria.fr

Laurent Lefèvre
Inria Avalon Team / LIP Laboratory
École Normale Supérieure de Lyon
46, allée d'Italie
69364 Lyon Cedex 07 - France
laurent.lefevre@inria.fr

Denis Trystram
LIG
University Grenoble Alpes
150, avenue du Torrent
38401 Saint Martin d'Hères
denis.trystram@imag.fr

Abstract—EcoIndex has been proposed to evaluate the absolute environmental performance of a given URL using a score ranging from 0 to 100 (the higher, the better). In this article, we make a critical analysis of the initial approach and propose alternatives that no longer calculate a plain score but allow the query to be situated among other queries. The generalized critiques come with statistics and rely on extensive experiments (first contribution). Then, we move on to low-cost Machine Learning (ML) approaches (second contribution) and a transition before obtaining our final results (third contribution). Our research aims to extend the initial idea of analytical computation, i.e., a relation between three variables, in the direction of algorithmic ML computations. The fourth contribution corresponds to a discussion on our implementation, available on a GitHub repository. Along with the paper, we invite the reader to examine the question: What attributes make sense for our problem?, or equivalently, what is a relevant data policy for studying digital environmental impacts? Beyond computational questions, it is important for the scientific community to focus on this question in particular. We currently promote using well-established ML techniques because of their potential, which we discuss in the paper. However, we also question techniques for their frugality or otherwise. Our data science project is still at the data exploration stage. We also want to encourage synergy between technical expertise and business knowledge because this is fundamental for advancing the data project.

Index Terms—Computing for Sustainability, Using Big Data Analytics to improve sustainability, Measuring the environmental impact of HTTP requests with ML techniques, and Challenges for an environmentally sustainable ICT Industry.

I. Introduction

To avoid dramatic consequences of climate change, all sectors of the global economy, including Information Communication Technology (ICT), must keep their greenhouse gas (GHG) emissions in line with the Paris Agreement [1]. The literature contains estimates of ICT's GHG emissions, which put ICT's share of global GHG emissions between 1.8% and 2.8% in 2022[1]. Authors in [2] found pronounced differences and many debates concerning the underlying assumptions behind the documented studies, which could suggest that global emissions from ICT are as high as 2.1%–3.9%. Some scenarios expect a rate of 20% in 2050 [3], which is considerable.

Can we quantify the environmental footprint when we click on a web page? To note the quality of an HTTP (HyperText Transfer Protocol) request, let's imagine that we have a tool like the Nutri-Score in France for food products that allows us to give it a grade between A and G. Let's also assume that these requests are archived in a public database like the HTTParchive[2]. By regularly exploring this database, we could track the evolution of the environmental footprint of Web requests. This article first presents the logic behind the EcoIndex environmental metric[3] and discusses its limitations.

Our motivation comes from our belief that the initial EcoIndex model is unsatisfactory (we call it the *historical* EcoIndex) and thus, we propose a new scientific direction to bypass certain limitations, for instance, if the methods scale well with an increased number of attributes. Thus, we make proposals to extend the initial definition according to new attributes, all of

[1]https://www.youtube.com/watch?v=UM3EW01_PUY&t=143s&ab_channel=Jean-MarcJancovici
[2]https://httparchive.org/
[3]https://www.ecoindex.fr/a-propos/

which relate to energy consumption or production and location. The idea is to switch from a purely analytical view of the EcoIndex metric to a 'Data Science' problem because of the potential for more investigations in that domain, for instance, in considering dimensionality reduction techniques.

We use basic and effective ML techniques to gain more confidence in the research direction. So, the objective is also to investigate the step of data exploration of any Data Science project while simultaneously questioning the nature and type of data to be considered in building a model that is acceptable and understandable to all. It's in this case that we could leverage mechanisms to change habits and get the public to adhere to an eco-responsible digital approach.

Thus, the paper organization is as follows: Section II discusses related works, tools, and techniques related to 'power' monitoring, generally, or for the HTTP. Section III presents the calculation of the historical EcoIndex and its limits. We also make a statistical study to show if our assumptions about limits are a reality. Section IV describes our approaches and positions them with historical EcoIndex. Each new approach has an experimental part to highlight the conceptual differences to the historical calculation. Section V gives the lessons learned and general comments. Finally, Section VI summarizes our work and presents some exploratory directions.

At last, note that, section III and section IV follow their own "methodology and results" format. We do prefer this organization to bring closer ideas and concrete results rather than a unique experimental section.

II. Related works

A. Impact of ICT on the Environment

New information technologies (ICT) are rapidly evolving and deployed to make our world smarter, safer, and more sustainable. Based on [4], the author distinguishes three levels of effects of ICT on the environment: 1st order effects (direct effects of ICT caused by their physical production, use, and disposal), 2nd order effects (impacts of ICT on other sectors), and 3rd order effects (structural ones), which include rebound effects [5], among them, the direct, indirect, and economy-wide effects. In the case of the direct rebound effect, lower energy cost induces price reductions that trigger an increase in the demand for the cheaper good. In the case of the indirect rebound effect, when a resource is used more efficiently, and its price goes down, it induces the consumption of other commodities (e.g., consumers buy extra TVs with the money they saved due to an energy-efficient washing machine or dishwasher of grade A+++). The economy-wide rebound effect appears when declining energy prices reduce the prices of intermediate and final goods throughout the economy and cause structural changes in production patterns and consumption habits. For example, cheaper gasoline enables people to live further away from their workplace by making it less expensive to drive longer distances to work.

Thus, it is crucial to examine technologies based on their potentially harmful effects on a sustainable world. We started such a project in [6], a preliminary work on the environmental impact of HTTP requests according to the EcoIndex vision, and where we put forward and test ideas. Contrary to this previous work, this paper provides more solid investigations and assumptions verification. It also contains more experimental work, available on GitHub to replay all the scenarios we discuss.

B. Estimating and measuring the energy consumption of IT pieces of equipment

Different techniques estimate power from hardware characteristics or power measures outside the hardware. Hardware performance counters help deliver cache misses and many other low-level details. They are connected to an API (Application Programming Interface) to expose their values to software power models. Several software-based power meters [7] are available to estimate power consumption and energy usage, while hardware power meters are external devices that can also measure power consumption.

The HPC PowerStack community promotes tools, among them the Variorum[4], an extensible, vendor-neutral library for exposing power and performance capabilities of low-level hardware events across diverse architectures in a user-friendly manner.

Regarding the measure of environmental indicators, people have also developed Python libraries such as CodeCarbon[5], CarbonTraker[6], and ImpactTracker [8]. CodeCarbon, which we use, in this paper, estimates and tracks carbon emissions from your compute engine and quantifies and analyzes their impact. CarbonTracker is a tool for tracking and predicting the energy consumption and carbon footprint of training deep learning models as described in [9]. ImpactTracker creates a leaderboard for energy-efficient reinforcement learning algorithms to incentivize responsible research in this area as an example for other areas of ML.

III. Presentation of the calculation of the historical EcoIndex and critical analysis

A. Predicting energy consumption of IT resources

Estimating the carbon footprint of human activities can only be done indirectly. The method used, in general, relies on a targeted activity model. This is

[4]https://variorum.readthedocs.io/en/latest/
[5]https://github.com/mlco2/codecarbon
[6]https://github.com/lfwa/carbontracker

the case for EcoIndex[7], which is only concerned with HTTP requests. This metric is based on the "3-tier" concept considering the three parameters client, server, and network, which in the model are weighted. In short, EcoIndex is a tool that, for a given URL, makes it possible to evaluate its absolute environmental performance using a score out of 100 (higher is better), its relative ecological performance using a score from A to G, the technical footprint of the page (weight, complexity, etc.), and the associated environmental footprint (greenhouse gases and water).

The "historical" version of EcoIndex is a plug-in to be installed on the browser that works as follows: One provides a URL to EcoIndex, which transfers it to the server side. The server responds and returns to the browser an HTML page containing the answers to the request. The plug-in measures the footprint of the application in terms of the number of elements in the web page (the number of HTML tags, noted DOM), the number of requests in the returned page (noted REQ), and finally, calculates the number of bytes of the returned HTML page (noted SIZE) that have passed through the network. Thus, EcoIndex implicitly takes into account the cost of sending data across the network. These values are fed into the EcoIndex algorithm to calculate the performance and environmental footprint.

B. How does EcoIndex work?

The term EcoIndex refers both to a set of best practices for building a website and to a software tool that evaluates several factors for a given URL:

- its absolute environmental efficiency using a score function on a scale of 0 to 100 (the higher the score, the better),
- its relative environmental performance using a score from A to G as known for household devices or food (Nutri-Score),
- the technical footprint of the page (weight, complexity, etc.),
- the associated environmental footprint (greenhouse gases generated, water resources consumed, etc.).

EcoIndex aims to help as many people as possible become aware of the environmental impact of HTTP requests and propose concrete solutions to reduce it. Let us now zoom in on the four previous metrics (which do not consider the best practices component). They are simple to understand, even for non-specialists. However, this model has limitations that will be explored step by step in the following.

C. Limitations inherent to the 3-tier model

The complementary analysis of an expert is essential for a complete and reliable operational evaluation of environmental performance. Indeed, EcoIndex does not take into account, in the sense of a life cycle analysis, the environmental impact of the computer making the request nor of the complete user's browsing. Only a query isolated from usage is analyzed, such as the Nutri-score or the washing machines purchased at any supermarket store. Similarly, when the request is resolved on the server side in a data center (for example, at Google when the URL is http://www.google.com), EcoIndex does not take into account the environmental impact of these servers in the classical sense of life cycle assessments (LCA), nor of the different network equipment that is crossed between the user terminal and the data center[8].

EcoIndex is not at the same level as an LCA, and it is a tool anchored in the 3-tier model, a high-level approach with limitations to capture fine-grained phenomena. However, it allows for discussion of models and their attributes that would significantly characterize the environmental impact of the Web, reduced to the dimension of HTTP requests. The positive sides of EcoIndex are that the loading, creation, and display of the page in the browser are not simulated and that the three parameters DOM, REQ, and SIZE (see Equation 1), capture an architecture that governs the macroscopic operation of a request on the Web thus EcoIndex makes sense.

D. Limitations inherent to the calculation itself

In detail, the environmental performance is calculated based on normalized parameters representing each third of the client/server architecture (see Equation 1) by a weighted sum, where the weights are determined by macroscopic studies once and for all. They do not consider variations over time or the user's geographical location.

Moreover, when looking closer at the open source implementation of EcoIndex, it is not directly the parameters DOM, REQ, and SIZE that are considered but values corresponding to quantiles. In Equation 1, quantiles are hidden in the F functions. A few values have been determined by retrieving the three parameters from URLs in a reference URL database, the HTTParchive[9]. This database offers open data, and, in our case, we used the April 2022 extraction. Basically, each line of the extraction gives a URL, i.e., a request that occurred in April 2022 on the Internet.

One question that arises is the stability of these quantiles over time: are they the same in 2023 as in 2020, when they were determined for the historical EcoIndex? A priori, websites are regularly reviewed to adopt better eco-design practices over time; therefore, there is no reason for the quantiles to be fixed once

[7]https://github.com/cnumr/GreenIT-Analysis

[8]https://ecoinfo.cnrs.fr/category/acv/

[9]https://httparchive.org/

$$EcoIndex = 100 - \frac{3 * F_{DOM}(DOM) + 2 * F_{REQ}(REQ) + 1 * F_{SIZE}(SIZE)}{6} \quad (1)$$

and for all. In the same vein, we can also make a minor remark on the fact that some sites, for example, those of the big media, are dynamic and that the value of EcoIndex is likely to evolve from day to day, but probably not too abruptly, for example from A to G. EcoIndex seems to us to be robust to this phenomenon. However, the A-G grades correspond to the EcoIndex ranges of 100-81 for A and 10-0 for G. How were these different limits determined? Do they compare to the quantiles for the HTTParchive EcoIndex measures? They are close but different.

E. Limitations inherent in the attributes that make sense

The historical model only lends itself, a priori, to introducing new attributes other than the 3-tiers in the model incompletely, as we have pointed out (not taking into account the environmental impact of the user's terminal and the server). We could consider, for example, introducing in the model notions of the energy mix and, in this case, propose a new EcoIndex+ indicator that would show grades turned to A for decarbonated energies used on the client and server-side and grades turned to G if the powers involved are carbonated. If the HTTP request goes through a 4/5G mobile, we could also aggregate the CO2 impact of the operator, which would lead to a richer view of the EcoIndex+.

To be more exhaustive in the attributes to be injected in EcoIndex+, we need, on the one hand, for the community to agree on these new attributes, and then we need to have associated computational methods able to process a large number of attributes using classical machine learning techniques that allow for processing models with several hundreds of characteristics at low cost!

F. Statistical analyses of our dataset

We first conducted some experiments for the historical EcoIndex to check if working with quantiles leads to issues or not. Experiments correspond to statistical treatments, and a dedicated Jupyter notebook is available on our GitHub repository[10]. We used a dataset of more than 100k URLs, all coming from the April 2022 dataset of the HTTParchive. We first observed the distribution of requests' number, DOM number, and transfer data size parameters. We noticed that the number of DOM nodes does not follow a Poisson distribution like the number of HTTP/HTTPS on the page or the number of kb transferred.

[10]https://github.com/christophe-cerin/Ecoindex-Revisited

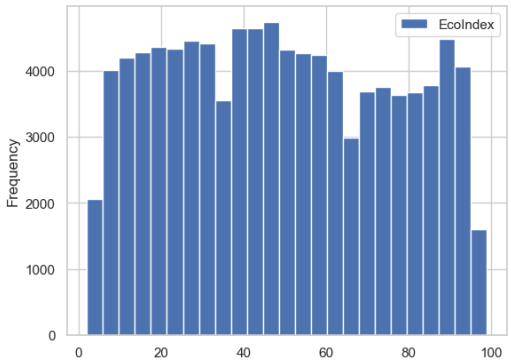

Figure 1. Distribution of EcoIndexes

However, we also considered the empirical distribution function (also called an empirical Cumulative Distribution Function, eCDF), which is the distribution function associated with the practical measure of a sample. Each value at any specified value of the measured variable (DOM, REQ, or SIZE) is the fraction of observations of the measured variable that are less than or equal to the specified value. In this case, we also noticed the Poisson distribution of the parameters, with a slight difference in the number of DOM nodes.

The ExoIndex has a uniform distribution on our dataset; see Figure 1 confirming that our dataset is diverse and big enough regarding the dataset initially used by GreenIT. There is nothing special to notice except that the numbers of values at the extreme part of the spectrum are few, which is expected and confirms that the definition of the EcoIndex is relevant.

Then we studied the impact of each parameter (DOM, REQ, SIZE) on the EcoIndex. The pair plot we used helps us understand the best features to explain a relationship between two variables or form separate clusters. The main diagonal subplots are the univariate histograms (distributions) for each attribute. Check with the notebook.

We noticed that the parameters are highly correlated visually. We also noticed that the parameters correlate highly with the EcoIndex, as was expected. While the number of DOM and the number of requests relate higher than 0.8, the data transfer size has a 0.63 correlation which indicates that it is less significant. If we include the outliers, the correlations drop between 0.3 and 0.6.

The original javascript code considers 10, 25, 40, 55, 70, and 80 as delimiters to give the grades (A-G). We decided to compute the quantiles for each parameter since we know their measured values for each HTTP request. Then, we noticed that the difference in values

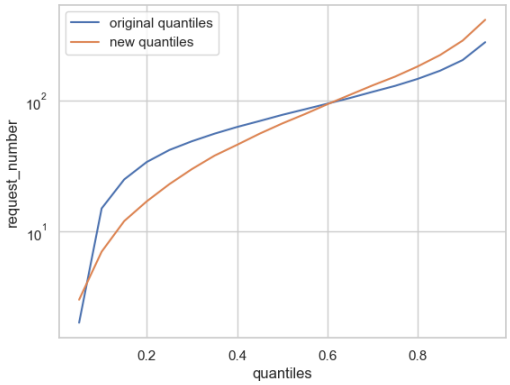

Figure 2. Quantile's similitude for the request parameter

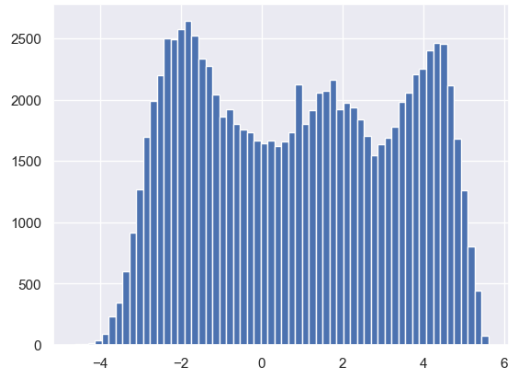

Figure 3. Histogram of the difference between quantiles

for the quantiles (the original and the computed ones) is relatively small. We observe in Figure 2 a larger difference for the request variable.

Then, we switched to the computation of the EcoIndex with new quantiles. The distribution of the new EcoIndex, as with the historical EcoIndex, follows that of the number of DOM nodes, and this is logical since it is the most impactful parameter. The two distributions obtained with new and historical quantiles are close. Intuitively, this point was unexpected, but it confirms a slight variation in the quantiles between 2020 (the ones used for the historical EcoIndex) and 2022 (the ones computed on our 2022 dataset).

In Figure 3, with 60 bins for the histogram, we observe that the difference in values for the two EcoIndex is at a maximum of 2500 on the y axis and that the difference may be positive or negative, ranging from -4 to +5.5 on the x. Thus, Figure 3 represents the distribution of the difference between calculations. Since the difference is slight, the two computed EcoIndex are similar. Finally, we found that the cosine similarity metric is close to 1, meaning that the computed EcoIndex (historical and new) are very similar. Thus, the new quantiles do not significantly impact the measurements. We conclude with a stability of the original quantiles over time, meaning that we can continue to use the quantiles from the historical EcoIndex.

IV. The novel approaches

A. Motivations and hypotheses

We showed in the previous section that the process does not require calculating each quantile again each time we change the dataset. However, we have not answered what happens if the weights change. In this case, it is, therefore, necessary to launch a benchmark, the HTTP archive, for example, at each new weighting. Remember that the choice for the weights is based on macro-studies of the ICT field, which is out of the scope of this paper.

We now discuss approaches that do not require a systematic switch to a benchmark; thus, they are more generic than the previously known ones. We consider two types of approaches. The first ones continue to produce a score, whereas the second builds a map of the global situation.

In the first case, we design approaches without the notion of quantiles in mind. In the second case, we can position a given HTML request regarding similar requests. These methods have been proposed in [6]. In this paper, we provide arguments to compare them from a qualitative point of view. We aim to study how robust one proposal is, compared to another. We also aim to experiment with categorical data and for the second type of approach.

B. Techniques from the data sciences

We can turn to data science techniques to satisfy the requirements that have just been stated. The techniques of the first type allow us to project a point in an N-dimensional space onto the line while preserving the desirable property. If two points are close in the N-dimensional space, they will also be nearby on the line. Thus, the EcoIndex remains associated with one and only one value: the position on the line is the EcoIndex measure. This measure is easy to implement and does not need to go through quantiles. The triplet (DOM, REQ, SIZE) is projected directly on the line. The potential bias of quantiles is avoided.

1) Techniques that reduced to a single value: We now give a brief overview of the techniques. Remind that implementations, we mean the full details, are available on our GitHub repository or in reading our preliminary work [6].

The first technique we use, according to this vision, to compute the EcoIndex is based on Random Projection [10], [11], [12] This well-known technique is a widely popular technique used in approximate similarity searches. Approximate search allows high-dimension exploration search. Thus, the technique allows dimensionality reduction, which is one new goal

Table I
RANDOM PROJECTION VERSUS COLLINEARITY

Average Root Mean Square Error:	27.01
Min Root Mean Square Error:	0.009
Max Root Mean Square Error:	79.45

Table II
RANDOM PROJECTION VERSUS LSH-KNN:

Average Root Mean Square Error:	26.94
Min Root Mean Square Error:	0.019
Max Root Mean Square Error:	71.84

of this paper, aiming to prepare the future of EcoIndexes. Note that, if we try several random matrices for projections, all do not lead to the same ordering on the line. The important fact here is to keep the same random matrix for all the experiments.

The second technique uses the Locality Sensitive Hashing (LSH) concept. LSH [13], [14] is again a widely popular technique used in approximate similarity searches, i.e., for nearest neighbor searches in high-dimensional spaces.

The general idea is to consider the neighbors of a given $(DOM, REQ, SIZE)$ triplet, then to summarize all of them through a single value (centroid) given a certain $(DOM', REQ', SIZE')$, and then approximate the EcoIndex as the sum $DOM' + REQ' + SIZE'$. According to our formulation, two distinct queries may have the same neighbors because they are close in the initial 3-D space. Thus, they may both have the same EcoIndex. This effect is intended because we want to focus on the similarities between queries.

The third technique is to search for collinearities as follows. In geometry, two or more points are said to be collinear if they lie on the same line. Hence, the collinear points are the set of points that lie on a single straight line.

Intuitively, we compute 'the most collinear' vectors to a given vector $(DOM, REQUEST, SIZE)$, and then we summarize all of them through a single value (centroid) given a particular $(DOM', REQ', SIZE')$. Finally, we approximate the EcoIndex as the sum $DOM' + REQ' + SIZE'$.

The root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and observed values. Tables I, II and III make pairwise comparisons of the techniques, according to the RMSE quality metric, and over more than 100k URLs from our dataset (`url_4ecoindex_dataset.csv`). Remind that we compare EcoIndex values ranging from 0 to 100.

As a local conclusion of this study, the new approaches seem significantly different since the RMSEs are sparse. We are in the presence of three different techniques that reduce the computation to a single

Table III
LSH-KNN VERSUS COLLINEARITY

Average Root Mean Square Error:	3.849
Min Root Mean Square Error:	0.009
Max Root Mean Square Error:	20.32

value, and the "best" technique to choose remains open. At this time of our project, we do not have strong arguments in favor or against one technique regarding the closest technique to the "ground truth" (the ideal expected result). This is still an open question. One step forward could be to analyze and compare the distribution of the EcoIndex values given by each method. Another possibility we explore further in the paper is to consider that the "ground truth" is given by the method with the less energetic consumption or the less CO2 production method.

We also conducted experiments to compare the RMSE of the historical EcoIndex with the three other methods, and the conclusion is similar to the previous one. Regarding the metric, the historical EcoIndex is different from all other methods.

However, a comparison between techniques can be made by considering, first, the execution time of one execution and, second, the carbon footprint to solve an HTTP request. CodeCarbon, as we mentioned earlier, is a Python software package that seamlessly integrates into your Python codebase. It estimates the amount of carbon dioxide (CO2) produced by personal computing resources to execute the code.

Table IV shows the measurements we have taken for the energy consumption and CO2 emission of the four methods, including the historical EcoIndex, referenced as *EcoIndex* in the table. Given 100k $(DOM, REQ, SIZE)$ triplets, we measured the environmental costs of the execution for computing the EcoIndex. The Energy and CO2 metrics are those given by the Codecarbon tool. The notebook was an 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz with 32GB of RAM.

It is worth noting that, as expected, because we use a KNN search with the LSH implementation, which demands a lot of resources, the LSH method is the most critical, having an order of magnitude even less good than the collinearity method, hence an X in the column. We also observe that the collinearity method is too energy inefficient because of the implementation that requires too many intermediary data structures. Finally, the best method is the random projection method, as expected, due to its algorithmic simplicity. This is the method of choice because it scales up (with respect to the number of parameters it can process) compared to the calculation of the historical EcoIndex.

Notice that our implementation of the historical EcoIndex is sensitive to the location where we

Table IV
ENERGY AND CO2 EMISSION FOR THE EXPLORATION OF 100K URLS

	EcoIndex	Random proj.	LSH	Collinearity
Energy	0.000121 kWh	0.000011 kWh	X	0.036015 kWh
CO2	5.806e-05 kg	2.676e-08 kg	X	0.0172348 kg

launch the request, hence a big difference in the CO2 emissions, even for requests that require few resources (grades A and C). Table V illustrates this particular point. The two first lines correspond to an EcoIndex computed from Tokyo inside the building of the National Institute of Informatics (NII) on two pages hosted by NII, and the last two lines correspond to an EcoIndex computed from France on the same two URLs. Obviously, it is the same notebook that carried out the four requests (CPU model: 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz). The only change in the experimental setting is the location for emitting the request we utilize. Notice that the performance is better from France than from Japan. The Eduroam network used in NII to access the NII web server performs poorly compared to access from outside NII. However, the important point to notice is the impact of the network on the determination of the emissions. The corresponding codes, available on our GitHub repository, for monitoring the energy and emissions are `codecarbon_collinearity.py`, `codecarbon_test_ecoindex.py`, `codecarbon_random_projection.py`, and `codecarbon_test_ecoindex_bis.py`.

Note also that the measures in Table V correspond to a measurement of the full cycle (determination of the three parameters, then the computation of the EcoIndex, given the parameters). In contrast, values in Table IV correspond to the computation of the EcoIndex only, given the parameters.

From the experiments and results shown in Table V, we conclude that it makes sense to consider the location (in some way to be determined) in the computation of an environmental impact of HTTP requests.

C. Building a map

Instead of reducing the problem to a single value, we propose to use the Self Organizing Map (SOM) representation. SOM is usually present in the exploration phase of a data science project, and it clusters data for finding representative data in large datasets. It is an unsupervised machine learning technique used to produce a low-dimensional (typically two-dimensional) representation of a higher-dimensional data set while preserving the topological structure of the data. SOM forms a semantic map where similar samples are mapped close together and dissimilar ones apart.

In practice, SOM usage is for clustering, data visualization, anomaly detection, etc. More concretely, we can use SOMs to build a recommender system or to identify trends and patterns. Another example, we can also draw a U-Matrix, representing a self-organizing map (SOM) where the Euclidean distance between the codebook vectors of neighboring neurons is depicted in a grayscale image. This image is used to visualize the data in a high-dimensional space using a 2D image.

In this subsection, we engage our study in the direction of dealing with much more than three attributes. The general question is, for instance, "How to introduce new attributes into the model to capture specific insights related to the environmental impact of ICT" as the energy mix. We would also like to consider if the HTTP request goes through a 4/5G network or fiber. We want to aggregate in the model the CO2 impact of the TELCO operator. In short, we are convinced that we have to deal with a potential of 10, 20, or more attributes, all related to “energy.”

We, therefore, propose to focus on the following attributes found in public datasets at Enedis[11] (France’s national electricity supplier) and Arcom[12] (France’s telecom regulatory agency), in addition to the $(DOM, REQ, SIZE)$ attributes.

From Enedis, we have access, city by city, to the annual consumption and production of electricity, which can be low-carbon or not. The Arcom performs throughput tests (4G/5G or fiber) based on HTTP requests. It also gives the GPS position of the location from which the HTTP request originated. So we can know the city and link it with Enedis data.

The intuitive idea is to introduce a new model based on the location and the type of electricity used. We do not believe that "a universal" grade, independent of the territory, makes sense. We also believe that making HTTP requests from a device powered by green energy should lead to a better overall score than a device powered by a non-green source.

Topological maps (SOM) help us in this respect. They group together in similar situations. So, by zooming in on the points on the map around a query, we’ll be able to find the cities that look like this query and the queries that look like me.

In this way, we no longer produce a grade but a fairly general view of the state of the System, having executed all the HTTP requests of the topological map. We retain the possibility of making comparisons for one or more attributes between two requests of our choice.

We have implemented this model according to Scikit-learn [15], and Susi [16] packages. Hyperparameters[13] for Susi are also documented, and we used the defaults. All the technical details (name and meaning

[11] https://www.enedis.fr

[12] https://www.arcom.fr/

[13] https://susi.readthedocs.io/en/latest/hyperparameters.html

Table V
SENSITIVITY ANALYSIS REGARDING THE LOCATION

Location	URL	GRADE	CO2
JAPAN	https://www.nii.ac.jp/en/faculty/digital_content/andres_frederic/	C	3.317e-05 kg
JAPAN	https://research.nii.ac.jp/~andres/official/content_e.html	A	3.154e-06 kg
FRANCE	https://www.nii.ac.jp/en/faculty/digital_content/andres_frederic/	C	1.220e-05 kg
FRANCE	https://research.nii.ac.jp/~andres/official/content_e.html	A	1.280e-06 kg

of attributes) are available on our GitHub repository. Finally, based on the `som_dataset.csv` dataset available on our repository, we get the topological map depicted in Figure 5. The $(DOM, REQ, SIZE)$ attributes are given by the EcoIndex method.

The attributes we consider in these experiments, namely those extracted by `som.py` and `som_test1.py`, are given in Figure 4. The corresponding CSV files are `som_dataset.csv` and `som1.csv`.

The view we get in Figure 5, based on `som_dataset.csv`, `som.py`, shows the distribution of the EcoIndex in one class (top of the Figure) and the distribution of the values into three classes (bottom of the Figure) according to the SOM clustering algorithm. Thus, now, we can query the classification to analyze the "points" with the same color and, maybe, to find patterns in our dataset of EcoIndex values. Here, for our toy example, namely the `som_dataset.csv` dataset, we found that the most impacting factor for the clustering is the request parameter.

The view we get in Figure 6, based on `som1.csv`, `som.py`, dealing with categorical data (operator, city, and URL), shows the U-matrix (unified distance matrix) where light colors depict closely spaced node codebook vectors and darker colors indicate more widely separated node codebook vectors. Thus, groups of light colors can be considered clusters, and the dark parts are the boundaries between the clusters. In this case, we observed that the data forms three clusters, which is more informative than the plot in Figure 7. With 7 it is more difficult to isolate the clusters in the top left and bottom right and the oblique one in the middle of the plot.

V. LESSONS LEARNED AND GENERAL COMMENTS

In this paper, we turned to data science techniques to satisfy the requirements stated and discussed in Section III. This scientific field has developed strategies to manage many attributes, such as random projection or Linear Sensitive Hashing (LSH) techniques. They are simple to understand, and this is positive to stay in the spirit of the simplicity of the original definition of the EcoIndex.

The first technique allows us to project a point in an N-dimensional space onto the line while preserving the desirable property. If two points are close in the N-dimensional space, they will also be close to the line. Thus, the EcoIndex remains associated with one and only one value. This measure is easy to implement and does not require quantiles like the historical EcoIndex. The potential bias of quantiles is avoided.

With our second technique, Data science also allows us to no longer define the EcoIndex metric as a reduction to one value but to consider a two-dimensional map obtained from a transformation of the N-dimensional space, where the points of this map are grouped by affinity. We can thus situate a particular triplet $(DOM, REQ, SIZE)$ to neighbors. The technique's name is Self-Organizing Map (SOM), and we developed an example with about 15 attributes.

Again, this technique bypasses the bias for quantiles and opens new directions for exploring datasets of the considered new domain. Our implementations and experiments aimed to validate that several alternative definitions to the historical definition lead to non-aberrant, acceptable, and differentiating EcoIndex values.

The competitive advantage of our methods is related to the fact that the chosen procedures, all of which come from data science, allow a scaling on the number of attributes we may consider now or in the future. Indeed, we exemplify that we should include in the definition of EcoIndex attributes of location (country, region, department, city, neighborhood) or even attributes of the type of electricity production (nuclear, gas, coal, solar, wind). The reason is twofold. First, we do not believe in a global grade. Second, we believe that local information is the more important for deriving a model. Third, the type of electricity powered by our computers should be introduced in the models that are supposed to capture the environmental impacts of ICT.

Technically speaking, in our implementations we encode attributes that are strings, such as city name, according to a hash, given an integer value.

Our methods would allow, by design, the number of attributes to increase while maintaining a fast computation time, thus saving energy. Thus, we are convinced that Data science is the right direction to go further. We also believe that we are on the right track for proposing recommendations in the long term. For instance, by examining the topological map (SOM)

```
Attributes selected in som_dataset by som.py:
'dom','size','requests','EcoIndex','GreenHouseGaz','water','PageChargeeMoins10s',
'temps_en_secondes','Conso totale (MWh)','Conso moyenne (MWh)','Photovoltaique'

Attributes selected in som_dataset by som_test1.py:
operateur;latitude;longitude;CP;ville;url;dom;req;size;EcoIndex;GreenHouseGaz;
water;PageChargeeMoins5s;temps_en_secondes;Conso_totale_(MWh);Conso_moyenne_(MWh);
Photovoltaique
```

Figure 4. Attributes under consideration with SOM experiments

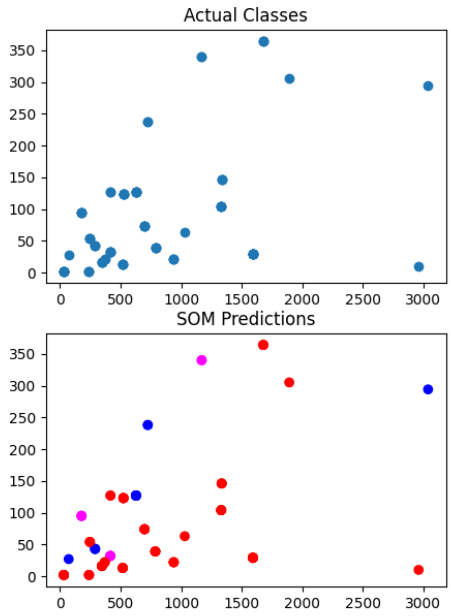

Figure 5. Topological map with public data from Enedis, Arcom issued from som_dataset.csv

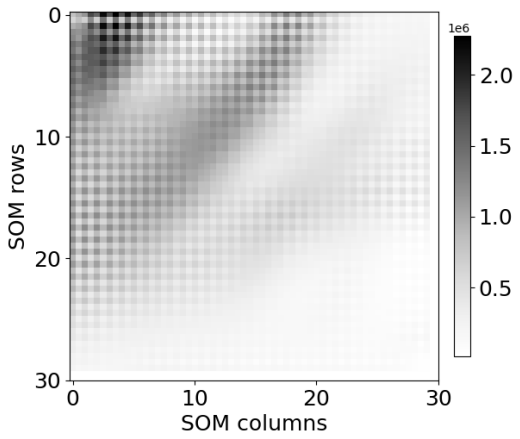

Figure 6. U-matrix with public data from Enedis, Arcom issued from som1.csv

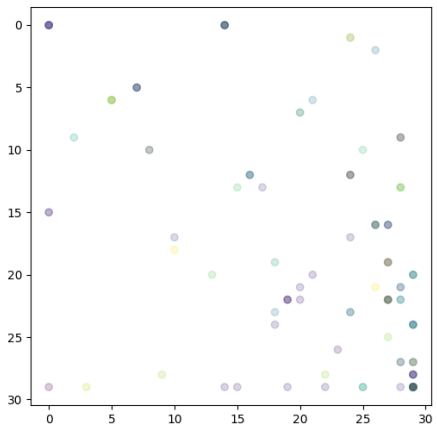

Figure 7. Clustering with public data from Enedis, Arcom issued from som1.csv

at a particular point, the recommendation could be to avoid visiting this or that website because it consumes or produces much more CO2 than another site. SOM provides richer information compared to a score. So now we come to the question of how to use the SOM, and for this, we need to turn to civil society and environmental impact experts.

VI. Conclusion and research directions

This paper discussed the EcoIndex, an environmental score for an HTTP request, and some of its limitations. It also compared qualitatively alternative approaches. We balance between interpretability and explainability with an insight into interpretability. Interpretability focuses on understanding the inner workings of the models, while explainability focuses on explaining the decisions made. On the explainability plan, we call on the entire IT community and the citizens to mobilize on the issues raised in this article. Crossing, in particular, competencies in Architecture, Systems, Networks, and Learning is necessary.

EcoIndex, from the point of view of environmental impact metrics and good practices in website ecodesign, is a simple approach that contributes to a better understanding of the phenomena and issues related to the role of digital technology in global warming. The

indicator is exciting in the logic of improving successive versions of websites.

We discussed several contributions: a detailed analysis of the weaknesses of the current calculation, consumption measurements for the new methods, the demonstration that location makes sense and that it should constitute one or more new attributes in the model, an implementation perspective of the most promising methods for a future and new measurement of the environmental impact of HTTP requests.

Much work remains to be done to deepen our knowledge and better understand the relationships between the different high-level models of the 3-tier architecture type and the field analyses of the life cycle of a digital product or equipment (LCA). On the other hand, it would be necessary to qualify and learn about data sets related to the studied domain (modeling phase) to use the model(s) in practice. Finally, there is the issue of standardization, particularly at the international level. Standards accompany the development and trade of products and services that meet objectives such as social concerns (consumer protection, safety, and security of services).

Thanks to the ideas presented in the article, the initial toolbox available on our GitHub repository for calculating the environmental cost is enriched with new methods from machine learning. In supplement to the dataset exploration steps, we propose now to investigate the question of utilizing the representations we exposed. As exposed above, we believe that a recommendation system for the citizens may help to encourage them to take a more environmentally friendly approach to ICT.

The work presented in this article is ultimately placed in an even more general context. First, can we observe, year after year, an evolution of the environmental impact of IT? Being optimistic, Web developers are concerned about good environmental practices and modify, over time, the HTTP pages they maintain having a bad EcoIndex. The Web, as a whole, has a chance to control its environmental impact for a long time from the moment we equip ourselves with a "weather forecasting system."

Acknowledgments

This work is partially supported by the Multidisciplinary Institute on Artificial Intelligence (MIAI) at Grenoble Alpes (ANR-19-P3IA-0003). This work is also conducted during the Délégation with Centre National de la Recherche Scientifique (CNRS) of Mr. Cérin. Thanks to the institutional support of the CNRS, the universities of Grenoble Alpes, and Sorbonne Paris Nord.

References

[1] UNFCCC, "The Paris Agreement," 2018. COP 21.

[2] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.

[3] The Shift Project, "Lean ICT: Towards Digital Sobriety," 2019.

[4] F. Berkhout and J. Hertin, "Impacts of information and communication technologies on environmental sustainability: Speculations and evidence," tech. rep., OECD report, University of Sussex, Brighton, vol 21, 2001.

[5] C. Gossart, "Rebound effects and ICT: A review of the literature," in *ICT Innovations for Sustainability* (L. M. Hilty and B. Aebischer, eds.), vol. 310 of *Advances in Intelligent Systems and Computing*, pp. 435–448, Springer, 2015.

[6] C. Cérin, D. Trystram, and T. Menouer, "The ecoindex metric, reviewed from the perspective of data science techniques," in *47th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2023, Torino, Italy, June 26-30, 2023* (H. Shahriar, Y. Teranishi, A. Cuzzocrea, M. Sharmin, D. Towey, A. K. M. J. A. Majumder, H. Kashiwazaki, J. Yang, M. Takemoto, N. Sakib, R. Banno, and S. I. Ahamed, eds.), pp. 1141–1146, IEEE, 2023.

[7] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters: focus on CPU and GPU," in *CCGrid 2023 - 23rd IEEE/ACM international symposium on cluster, cloud and internet computing*, (Bangalore, India), pp. 1–13, IEEE, May 2023.

[8] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," 2020.

[9] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," 2020.

[10] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, (New York, NY, USA), p. 245–250, Association for Computing Machinery, 2001.

[11] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, (New York, NY, USA), p. 274–281, Association for Computing Machinery, 2001.

[12] G. Siddharth, R.; Aghila, "Randpro- a practical implementation of random projection-based feature extraction for high dimensional multivariate data analysis in r." SoftwareX. 12: 100629. Bibcode:2020SoftX..1200629S. doi:10.1016/j.softx.2020.100629, 2020.

[13] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, (New York, NY, USA), p. 604–613, Association for Computing Machinery, 1998.

[14] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, (San Francisco, CA, USA), p. 518–529, Morgan Kaufmann Publishers Inc., 1999.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] F. M. Riese, S. Keller, and S. Hinz, "Supervised and Semi-Supervised Self-Organizing Maps for Regression and Classification Focusing on Hyperspectral Data," *Remote Sensing*, vol. 12, no. 1, 2020.