CarbonTag: A browser-based method for approximating energy consumption of online ads

José González Cabañas
UC3M-Santander IBiDat
Universidad Carlos III Madrid
Spain
jgcabana@inst.uc3m.es

Patricia Callejo
Universidad Carlos III Madrid
UC3M-Santander IBiDat
Spain
patricia.callejo@uc3m.es

Rubén Cuevas
Universidad Carlos III Madrid
UC3M-Santander IBiDat
Spain
rcuevas@it.uc3m.es

Steffen Svartberg
Cavai
Norway
steffen@cavai.com

Tommy Torjesen
Cavai
Norway
tommy@cavai.com

Ángel Cuevas
Universidad Carlos III Madrid
UC3M-Santander IBiDat
Spain
acrumin@it.uc3m.es

Antonio Pastor
Cavai
Spain
antonio.pastor@cavai.com

Mikko Kotila
Cavai
Finland
mailme@mikkokotila.com

Abstract

Energy is today the most critical environmental challenge. The amount of carbon emissions contributing to climate change is significantly influenced by both the production and consumption of energy. Measuring and reducing the energy consumption of services is a crucial step toward reducing adverse environmental effects caused by carbon emissions. Millions of websites rely on online advertisements to generate revenue, with most websites earning most or all of their revenues from ads. As a result, hundreds of billions of online ads are delivered daily to internet users to be rendered in their browsers. Both the delivery and rendering of each ad consume energy. This study investigates how much energy online ads use and offers a way for predicting it as part of rendering the ad. To the best of the authors' knowledge, this is the first study to calculate the energy usage of single advertisements. Our research further introduces different levels of consumption by which online ads can be classified based on energy efficiency. This classification will allow advertisers to add energy efficiency metrics and optimize campaigns towards consuming less possible.

Keywords: ad, online advertising, energy consumption, carbon emissions, sustainability

1 Introduction

The vast amount of energy consumed by human society is one of the biggest challenges faced by humanity today and in the near future. The production and consumption of energy contribute significantly to the emissions of greenhouse gases, including carbon, which contributes to climate change. The Statistical Review of World Energy 2022 by BP [1] shows that the primary energy demand increased by 5.8% in 2021, topping 2019 levels by 1.3%. This indicates that the global need for energy is continuing to rise. The International Energy Outlook 2021 [2] predicts that over the next 30 years, the world's energy demand will rise by roughly 50%.

While there is a growing interest in using clean renewable energy (the amount of renewable energy consumption growing by over 2000 TWh between 2019 and 2021), polluting fossil fuels consumption remained largely unaltered [1] and they are expected to be used beyond the decade 2050 [2], at which point the renewable energy is expected to become the most representative source of energy [3].

The transition from fossil fuels to clean energy is progressing more slowly than is necessary. It is paramount to implement strategies focused on reducing energy consumption to limit and prevent adverse environmental, social, and economic impacts, such as those associated with rising temperatures, rising sea levels, and decreasing precipitation. In addition, these strategies have recently become an immediate necessity in Europe, and the whole world, due to the energy crisis aggravated by the Ukraine War in 2022.

Since the advent of the internet in the early 1990s, the Information and Communication Technologies (ICT) sector has become a major driver of developed economies as well as a factor contributing to the increase in global energy consumption. Depending on the source, the total energy consumption of the Internet is approximated to be between 5% and 15% of total global energy consumption, that's is around 416.2 TWh per year [4–7]. This percentage is projected to rise sharply in the coming years.

Online advertising is one of the main sources of revenue supporting the operation of the commercial Internet. Millions of websites, social media platforms, mobile apps, video

platforms, etc, rely on online advertisements as their main source of revenue. As a result, each day, hundreds of billions of online ads are delivered to Internet users to be rendered by their devices. Indeed, online advertising is a $450 billion per year industry with an inter-annual growth of 35.4% between 2020 and 2021, according to the IAB's most recent Internet Advertising Revenue Report of 2021 [8]. This marks the industry's highest growth since 2006 and highlights the importance of online ads as a fundamental driver of the Internet-age economy.

The online advertising industry has developed a complex technology ecosystem, known as *ad tech* ecosystem, which is able to deliver personalized ads in nearly real time. This process may involve, for each ad delivery transaction, up to hundreds of different players with their associated data communications and processing events.

Previous literature has used high-level approaches to estimate that online advertising is a relevant contributor to the overall energy consumption of the Internet [9–11]. Therefore, without a doubt, online advertising must contribute to reducing the Internet's energy consumption by adopting more energy-efficient and therefore sustainable practices. The first step towards this consists in *defining accurate measurement techniques able to measure the energy consumption of the ad delivery process, if possible, at the granularity of individual ads.*

The complexity of this goal, requires, in order to succeed, to employ a *divide and conquer* approach. To this end, we have divided the ad delivery process into two separate parts: the *network* and the *device*. The *network* part involves all the data processing and communications among intermediaries participating in the delivery of ads, whereas the *device* part is defined by the ad rendering process executed by software installed in the device such as web browsers or mobile apps.

In this paper, we focus on the *device*, thus *our goal is to define an accurate measurement technique able to estimate the energy consumption of the rendering process of individual ads.*[1] We left the *network* part for future research. We conjecture the *network* part may be responsible for an importantly larger fraction of the overall energy consumption of the ad delivery process. However, properly measuring the *device* part is relevant since (as we will see) the aggregated energy consumption of the ad rendering process across hundreds of billions of ads each day is not negligible. Moreover, it will serve to create awareness among both advertisers and Internet users, which are the fundamental players fueling the online advertising ecosystem.

The goal of our paper is to define a methodology that allows measuring the energy consumption of the ad rendering process in real time and at scale. Ideally, one would wish to define a technique that measures the exact energy consumed resulting from executing the ad rendering process. Unfortunately, directly measuring the energy consumed by software processes such as ad rendering is not possible in practice (See Section 2). Instead, the alternative is to define techniques to produce accurate estimations of the energy consumption of the process. To this end in this paper, we present the two following contributions that allow providing an accurate estimation of the energy consumption of the ad rendering process in real time and at scale:

- First, we have run controlled lab experiments to measure the energy consumption of the rendering process for more than 25k different ads using three device configurations that combine different hardware and Operating Systems (OS). The obtained results represent to the best of our knowledge the largest ground-truth dataset of individual ads' energy consumption (See Section 3). We have leveraged this dataset to train a Machine Learning (ML) model that estimates the energy consumption of an individual ad from measurable parameters during the rendering process of the ad, such as: the Initiator Types, mean duration of TCP connections, duration of the ad rendering process, screen size, etc. Our ML model provides a good performance in the estimation of the energy consumption with R^2 values between 0,67 and 0,97 depending on the device configuration (See Section 4).
- Second, armed with our ML model, we have built CarbonTag(See Section 6). This system measures the level of energy consumption of the ad rendering process in real time and at scale. CarbonTagis formed by two components: an ad tagand a backend server. On the one hand, the ad tag(i.e., a JavaScript code) can be embedded by the advertiser or any intermediary in the ad. This ad tagis responsible for collecting the parameters required to feed our ML model and send them to the backend server. The backend server would execute the ML model, obtain an estimation of the energy consumption of the ad, record it, and optionally, respond to the ad tagwith such value. For instance, the ad tagcan use such information to show an energy label for the ad. Indeed, we propose our own energy labeling system inspired by the EU energy labeling framework. We have conducted extensive lab experiments to evaluate how CarbonTagoperates in real time and can scale to handle hundreds of billions of requests per day with minimal additional cost. CarbonTagis readily available to be integrated by advertisers and ad techintermediaries.

2 Background

2.1 Online Ad Delivery and Rendering Process

The current online advertising ecosystem implements what is referred to as *programmatic advertising*. When a user reaches a webpage (or mobile app), each ad space available on such webpage initiates the ad delivery process. The ad space sends an ad request message to the publisher's ad server requesting an ad to be placed in the ad space. The

[1]Note that in the remainder of the paper when we refer to ad energy consumption we are implicitly referring to the ad rendering process occurring in the device.

publisher's ad server can respond with a URL from where to receive the ad if a direct deal with an advertiser exists. Otherwise, the ad server forwards the ad request to an intermediary, which can be either an *ad network* or a *Supply Side Platform (SSP)*. The publisher's website and ad server, along with the SSP or ad network [2] form what is referred to as the *Sell Side* in programmatic advertising, since they are the entities involved in selling ad spaces. The term SSP is oftentimes interchangeable with the term Ad Exchange (AdX). Using the information included in the ad request message *bid request*, such sell-side actors process messages as described in the OpenRTB standard [12]. The *bid request* message is then sent to several *Demand Side Platforms (DSPs)* initiating a real time auction process. DSPs are technology platforms where advertisers (or their agencies) create, manage, and deliver ad campaigns. A DSP would check the information included in the received bid request, e.g., location of the device, profile of the user (age, gender, interests), type of ad space (banner, video, etc.), venue of the ad space (web domain or mobile app), etc. If these parameters match the criteria defined in any ad campaigns on the DSP, the DSP responds with a *bid response* including a bidding price for the ad space. The AdX runs a separate auction among the received bid responses and chooses a winner. The winner provides AdX with the URL of the ad to be delivered, which is hosted in the advertiser or DSP's ad server. This URL is forwarded back through the AdX, SSP/ad network, and publisher's ad server to the browser (or mobile app), which will render the ad.

The described process defines the *network* part of the ad delivery process. In addition, there are other subsidiary processes such as the tracking&profiling process. The ad techhas developed a sophisticated ecosystem to track users' activity online (but also offline, for example tracking users' locations), enabling the profiling of users. The profiles are provided to SSPs, AdXs and/or DSPs, which enrich the information included in ad requests and/or bid requests, allowing the delivery of personalized ads.

Once the browser (or mobile app) receives the URL of the ad, the *device* part of the ad delivery process starts (i.e., the ad rendering). The browser (or mobile app) download the ad, which is placed in the ad space, typically embedded in an iFrame (or even a double iFrame). The ad typically includes JavaScript code that is used for multiple purposes, including reporting delivery, monitoring user's actions (e.g., clicks on the ad), measuring KPIs (e.g., ad viewability [13, 14]) or identifying fraudulent practices [15, 16].

In this paper, we focus on measuring the energy consumption of the *device* part, i.e., the ad rendering process.

[2]Note that several ad networks or SSPs can be involved in the ad delivery process.

2.2 Measuring Energy Consumption of Digital Devices

Digital devices are, in essence, a set of hardware components used to execute software. The energy is consumed by hardware components, which directly depends on the demand of resources of the executed software, e.g., software requiring 100% use of CPU leads to higher energy consumption than software using 10% of CPU.

There are two main methods to measure the energy consumed by a device or by specific software within a device. The first, and most accurate one, is using an energy meter tool [17]. These tools measure the actual energy consumed by hardware devices. Therefore, if one can configure a setup where the device runs the software to be measured exclusively, the energy consumption measured by the meter is the energy consumed by such software. However, this method is not recommended to be used for software measurements. When software execution cannot be isolated, a common alternative is to measure the difference in the energy consumed when the device operates running and not running the software. The difference in the measured energy is considered a good approximation to the energy consumed by the software under analysis.

However, energy meters are not always practical. They have a minimum resolution, so energy consumption below that resolution cannot be measured. For instance, if the resolution of an energy meter is 10^{-3} Watts, then any software consuming less than this cannot be measured with this meter. In these cases, there is a commonly used estimation method: the number of CPU instructions executed by the software is counted [18]. The chipset manufacturer datasheet provides detailed information with respect to the consumption of energy of the CPU chipset under different conditions (e.g., % of the use of the CPU). Hence, an estimation of the energy consumed by a device in executing a given software can be obtained. Note that this approach assumes that the CPU is the hardware element consuming most of the energy in a device. The research community has developed open-source libraries that implement this method, such as CodeCarbon [19].

3 Dataset

The first task of this paper is to define an accurate model of the energy consumption by the rendering process of an individual ad. To this end, we need to create a ground-truth dataset providing the energy consumed by real ads rendered on websites.

We gathered two different sets of ads. The first one serves to create the model of the energy consumption of ad rendering processes. It is collected using an automated process that surfs the web to gather thousands of different advertisements. We refer to it as *automated* dataset.

To validate the generality of the model obtained from the *automated* dataset, we obtain a second dataset formed by ads shown to real users while they browse the web. We refer to this dataset as *human* dataset.

The remainder of the section describes the data collection procedure and our final datasets in detail.

3.1 Automated Dataset

To obtain a large number of online ads, we created an automated methodology that navigates through different websites and collects the ads present on the sites. This pool of ads will be utilized later to emulate the rendering of online ads and measure their energy consumption. There are four steps to the collection procedure:

(i) First, we compiled a list of websites to look for advertisements on. We used the Alexa list of the top 500 news websites and the top 500 most visited websites worldwide by Domain Authority [20].
(ii) Second, we require these websites to start displaying ads which we then can collect, so we implemented a methodology that automatically inspects the presence of cookie consent notices and accepts them. After accepting the consent notices, ads start appearing. The automated process then scrolls down the page five times (emulating the behaviour a user may have) to identify the presence of more ads on the webpage.
(iii) To retrieve the ads, our automated software uses a tool that can extract the HMTL from iFrames where ads are placed and save it to a file for offline analysis. We created a browser extension using as reference the extension eyeWnder [21] for this purpose. In particular, the extension analyzes the content of an HTML page, locates the iFrames that correspond to advertisements, and extracts their HTML code.
(iv) Having the browser extension, the navigation methodology, and the list of URLs to explore, we implemented the whole routine using the headless browser Selenium [22] and save each of the HTML from each ad in a separate file.

We created the *automated* dataset using the aforementioned procedure. It consists of a collection of 25557 ads, which serves to define the model of the energy consumption of individual ad rendering processes described in Section 4.

3.2 Human Dataset

The energy consumption model described in Section 4 is obtained using our *automated* dataset. These ads are obtained with automated software and thus, we have no guarantees that our model is actually representative of the energy consumption of ads shown to real users. To validate the generality of our model, we have then collected a second dataset including ads shown to real users (friends and colleagues), who have voluntarily installed a browser extension in their browser that captures the ads delivered. The *human* dataset is formed by 598 ads. Figure 1 shows the distribution of ad providers behind each ad of the *automated* and *human* datasets.

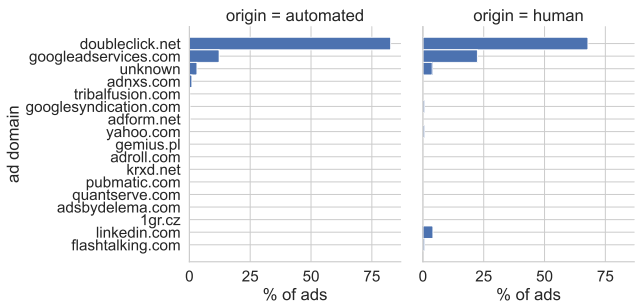

Figure 1. Distribution of ad providers from the list of 25k collected ads.

4 Data-Driven Model of the Energy Consumption of Ad Rendering Process

The cornerstone piece of our system is a model, which uses parameters available upon the ad being rendered on the user's browser to estimate the energy consumption of the ad rendering process in real time.

In this section, we thoroughly describe our methods for defining such a model. We first describe the extensive lab experiments conducted to create a ground-truth sample of ads' energy consumption using our *automated* dataset, which will serve to train and validate our model. Afterward, we provide a detailed explanation of the process we follow to come up with our final model.

4.1 Measuring Energy Consumption of an Ad

The first step to build our model is to obtain a measure of the energy consumption of thousands of online ads' rendering processes, which will serve as samples of the dependent variable of our model. In the remainder of the section, we describe the procedure used to obtain an accurate estimation of the energy consumption of the ad rendering process of the 25k ads from our *automated* dataset.

4.1.1 CodeCarbon Energy Measurement Tool. The online ad rendering process is a light software process that consumes in the order of 10^{-6} KWh.[3] As discussed in Section 2, commercial energy meters are unable to measure energy consumption with this granularity. Hence, following the recommendations of the literature, we opt for using a library that estimates the energy consumption based on the CPU cycles used by the software process. In particular, we use CodeCarbon [19], an open-source Python library that computes the energy and carbon footprints of any piece of code

[3]Note that this number may vary between a few 10^{-7} kWh and a few 10^{-6} kWh depending on the device where the ad is being rendered.

in various computer systems. CodeCarbon supports a variety of computing platforms, bases its estimations on the specific energy consumption reported by manufacturers' datasheets for tens of most commonly used CPU chipsets, and provides real time measurements. It calculates the amount of energy consumed by taking into account the computer architecture, usage, and operating time. The described features make CodeCarbon an appropriate tool for achieving our goal.

Moreover, CodeCarbon is able to identify the location of a device (e.g., using the IP address) and then convert energy consumption values into carbon emission values using national grid mix data based on the country where the device is located.

4.1.2 Lab Experiments to Measure the Energy Consumption of Ads.

We first present our lab infrastructure setup and then we describe the method to isolate the ad rendering process and measure the energy consumed by it.

Lab infrastructure set up. We set up a lab environment where each advertisement's energy consumption was measured using three real-world devices: a Windows desktop PC, a Linux desktop PC, and a Linux laptop. The only distinction between our two identically equipped desktop PC systems is their operating system (OS). Although the third device has different hardware than the first two, it runs the same operating system (OS) as one of them. Furthermore, every device is connected to the same network. As a result, three operating systems are available: a 64-bit Intel Windows 10 OS with 128GB RAM (Windows PC) with Google Chrome 103.0.5060.114, a 64-bit Intel Ubuntu 18.04 OS with 128GB RAM (Ubuntu PC) with Google Chrome 103.0.5060.114, and a 64-bit Intel Ubuntu 18.04 OS with 8GB RAM (Ubuntu Laptop) with Google Chrome 94.0.4606. The utilization of these different configurations aims to minimize the dependence of our final model on specific operating systems or devices. Moreover, we obtained 2 more devices for validating our model: a Mac laptop and a Windows Laptop. Our purpose is to build an ML model that later works on different unseen devices for the model and, therefore, validate it against real ads rendering in new different devices and operating systems. In conclusion, our devices for validation consist of the three mentioned above plus an HP Laptop 64-bit Intel Core i7-1065G7 Windows 10 Home OS with 12GB RAM (Windows Laptop) and Google Chrome 106.0.5249.61 and a Intel Core i5 Macbook Pro macOS Catalina 10.15.7 with 16GB RAM (Mac Laptop) and Google Chrome 105.0.5195.125.

Methodology to isolate the ad rendering process. We devised an automated method for isolating and running the rendering of each ad for each device. A diagram of the methodology is depicted in Figure 2. In the following we describe in detail the steps taken for measuring the energy consumption of the rendering process of an individual advertisement:

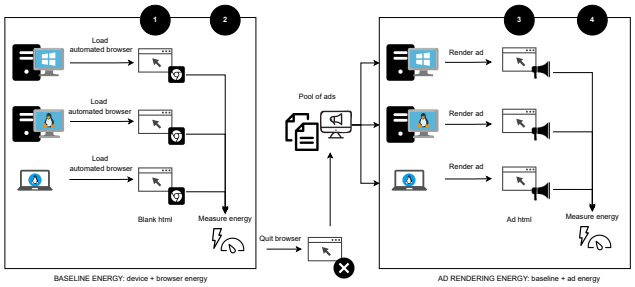

Figure 2. Diagram of the automated methodology implemented to measure the energy consumption of online ads.

1. The procedure automatically launches a new Google Chrome instance and opens a blank HTML file within it. The purpose of this step (step 1 in Figure 2) is to obtain the baseline energy usage before loading the ad.
2. Using CodeCarbon, we obtain the energy usage of the empty instance of Google Chrome with a blank HTML. Hence, this is referred to as *baseline energy* (step 2 in Figure 2). Immediately after, we quit the browser instance and we are ready to perform again the measurement but this time loading the ad.
3. After terminating the current Google Chrome instance, the process selects one ad from the *automated* dataset and launches a browser instance while rendering the advertisement. Then it waits until the ad is fully loaded.
4. Then, CodeCarbon measures the energy consumption (step 1 in Figure 2). This measurement includes the device plus the browser consumption together with the ad's consumption. We refer to this measurement as *ad rendering energy* since it measures the baseline consumption as in steps 1 and 2, together with the ad rendering process.

In summary, the described lab experiment provides two energy measurements for an individual advertisement: *baseline energy*, which measures the device and browser's energy consumption, and *ad rendering energy*, which measures the device, browser, and advertisements' energy consumption. We can use these two measures to define the dependent variable of our machine learning model as described in Section 4.2.1 (See Equation 2).

We repeat the process outlined above for each of the 25k ads in our *automated* dataset and each of the 598 ads in our *human* dataset. Furthermore, the procedure is repeated five times per ad to ensure consistent measurements, resulting in each ad measurement having five samples for *baseline energy* and *ad rendering energy*. Later, the median amount of energy used across these five samples is calculated. It should be noted that obtaining the energy consumption of advertisements through the use of this methodology takes time and is not immediate. On the slowest of the devices, the

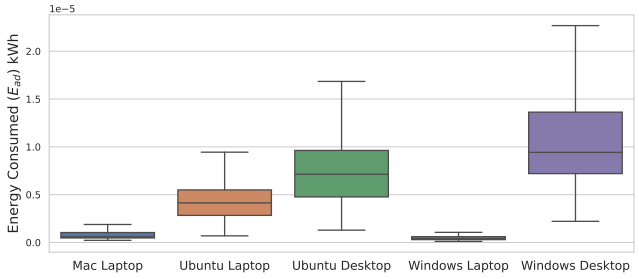

Figure 3. Energy Delta of each *human* ad for the different devices analyzed.

Windows PC, each ad measurement (i.e., the technique described above) took around 15 seconds. This means it would take us more than 4 days to collect the energy consumption measurements for the 25k advertisements. We repeated the measurements five times for accuracy and consistency, which took about 20 days of computation.

Note that during our lab experiments, for each of the 25k ads we also collect the parameters that will serve as independent variables in our ML model. Therefore, our experiments provide all the required input data for our model to estimate the energy consumption of the ad rendering process. As in the case of the energy, we collected 5 samples of the value for each parameter and consider its median value for building our model.

4.2 Building a Model to Estimate Energy Consumption of Ads

Once we have measured the energy consumption of the 25k advertisements in our *automated* dataset, we intend to develop an ad energy consumption estimator that can forecast the consumption of each ad based solely on the collected parameters during the ad rendering process. This model enables the possibility of estimating the energy consumed by each ad in real time as the advertisement is rendered, which is our main goal.

In the following, we go over the models, features, and the used ad parameters in great detail.

4.2.1 Metric to Build the Estimator. Before continuing with the description of our model, we first introduce *Ad Energy* (E_{ad}), a metric that directly reports the energy consumed by the rendering process of an ad. This metric is computed using the *baseline energy* and *ad rendering energy* described above as follows:

$$E_{ad} = ad\,rendering\,energy - baseline\,energy \quad (1)$$

Using *Ad Energy* (E_{ad}), we can report the actual ad energy consumption within the device on which the ad is running. This is to say, the actual energy (kWh) consumed by the advertisement rendering process on that specific device.

Figure 3 depicts a boxplot showing the energy consumed by the rendering process of each ad from the *human* dataset in each device used in our lab test environment. We can observe that the standard energy used by a certain hardware+software system differs from another hardware+software combination. The hardware in a laptop, which is designed to use less energy, is different from that in a desktop which is designed to be powered by an external source. For this reason, we determined that it is necessary to measure the relative energy consumption of an advertisement rather than the actual energy consumption (E_{ad}), which depends on the hardware to be measured. In order to do that, we compute the advertisement's relative energy usage, irrespective of the hardware on which it runs. Therefore, if we have two ads, *A* and *B*, we will not know exactly how much energy each of them uses (since it will vary depending on the hardware), but we will know that A uses more energy, and generally how much more energy, than B.

Hence we present *Normalized Ad Energy* (nE_{ad}), a metric that informs about the relative use of energy of the ad rendering processes with respect to the baseline energy consumption of such device:

$$nE_{ad} = \frac{ad\,rendering\,energy - baseline\,energy}{baseline\,energy} \quad (2)$$

Normalized Ad Energy acts as the dependent variable (i.e., the variable to be predicted) for our model.

4.2.2 Feature Selection. As indicated above, our lab experiment will collect for the 25k ads from our *automated* and *human* datasets the list of parameters described in Table 1, which are the initial set of independent variables under consideration to be used in our model. We undergo a feature selection process, to make the final selection of this initial set of parameters that will be used in the model.

To better understand our data and to discard less relevant independent parameters, we first made a manual inspection to see the distribution of each parameter, outliers, and the correlation with the dependent variable, i.e., the normalized energy consumption. This process suggested that the following independent variables should be included: screen_size, totalJSHeapSize, entries, et_paint, et_resource, it_xmlhttprequest, it_img, it_script, ad_navigation_duration, ad_navigation_processing, ad_navigation_onLoad, duration_mean, redirectTime_mean, request_mean, and response_mean.

Moreover, before developing our model, we performed an automatic feature selection process on these pre-selected independent variables to prevent over-fitting and so improve the performance of our estimator.

First, we keep the independent variables showing the highest correlation with the dependent variable (nE_{ad}), filtering out those variables with a correlation < 0.8.

Parameter	Description
usedJSHeapSize	The currently active segment of JS heap, in bytes.
totalJSHeapSize	The total allocated heap size, in bytes.
entries	Number of PerformanceEntry objects in the ad
entries_requested	Number of PerformanceEntry objects in the ad with a performance duration >0
screen_size	Screen width x screen height
et_element	Number of element entryType entries executed in the ad
et_navigation	Number of navigation entryType entries executed in the ad
et_resource	Number of resource entryType entries executed in the ad
et_mark	Number of mark entryType entries executed in the ad
et_measure	Number of measure entryType entries executed in the ad
et_paint	Number of paint entryType entries executed in the ad
et_longtask	Number of longtask entries executed in the ad
it_element	Number of element initiatorType entries executed in the ad
it_css	Number of css initiatorType entries executed in the ad
it_embed	Number of embed initiatorType entries executed in the ad
it_img	Number of img initiatorType entries executed in the ad
it_link	Number of link initiatorType entries executed in the ad
it_object	Number of object initiatorType entries executed in the ad
it_script	Number of script initiatorType entries executed in the ad
it_subdocument	Number of subdocument initiatorType entries executed in the ad
it_svg	Number of svg initiatorType entries executed in the ad
it_xmlhttprequest	Number of xmlhttprequest initiatorType entries executed in the ad
it_navigation	Number of navigation initiatorType entries executed in the ad
it_other	Number of other initiatorType entries executed in the ad
duration_mean	mean duration of the performance entries in ms
transferSize_mean	mean transfer size of the performance entries of the fetched resource
dedodedBodySize_mean	mean transfer size of the performance entries received from the fetch (HTTP or cache) of the message body
redirectTime_mean	mean duration of the redirect phase of entries in ms
app_cache_mean	mean duration of the app cache phase of entries in ms
dns_mean	mean duration of the dns phase of entries in ms
tcp_mean	mean duration of the tcp phase of entries in ms
request_mean	mean duration of the request phase of entries in ms
response_mean	mean duration of the response phase of entries in ms
ad_navigation_duration	Time in ms of the ad's execution
ad_navigation_transferSize	A number representing the size of the ad
ad_navigation_decodedBodySize	A number that is the size (in octets) received from the fetch (HTTP or cache) of the message body
ad_navigation_app_cache	Duration of the APP CACHE phase of the ad in ms
ad_navigation_dns	Duration of the DNS phase of the ad in ms
ad_navigation_tcp	Duration of the TCP phase of the ad in ms
ad_navigation_request	Duration of the REQUEST phase of the ad in ms
ad_navigation_response	Duration of the RESPONSE phase of the ad in ms
ad_navigation_processing	Duration of the PROCESSING phase of the ad in ms
ad_navigation_onLoad	Duration of the ON LOAD phase of the ad in ms

Table 1. Variables collected of each ad for analysing and populating the models.

variable	VIF value
ad_navigation_duration screen_size request_mean	5.78
ad_navigation_duration ad_navigation_onLoad	1.17
response_mean screen_size	5.12
ad_navigation_duration redirectTime_mean	2.94
ad_navigation_duration	4.32
screen_size	0.60
tcp_mean	1.44
request_mean	2.94
response_mean	8.42
it_xmlhttprequest	1.04
redirectTime_mean	2.77

Table 2. Variables used to build the ads energy estimator based on the information collected when simulating the rendering phase of each ad.

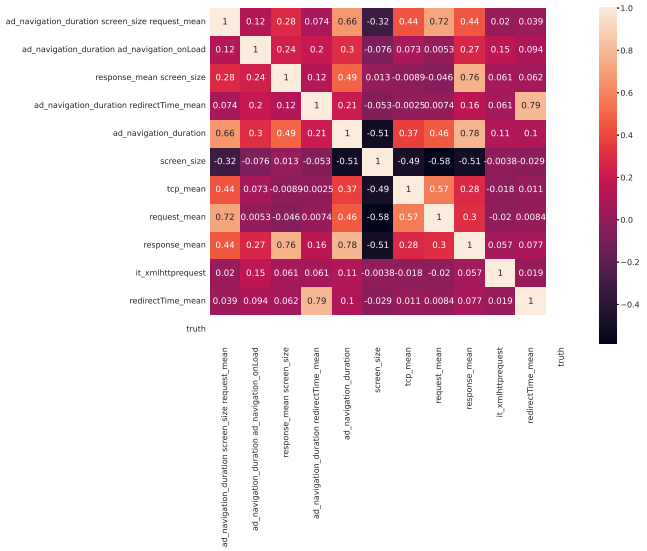

Figure 4. Correlations between each pair of independent variables used in the model.

Later, we quantify the multicollinearity using Variance Inflation Factors (VIF) to avoid considering independent variables with a high correlation with one another. We eliminate the variables with a VIF > 10, which indicates strong multicollinearity. [23]

Furthermore, because predicting energy consumption can benefit from the interaction of two or more variables, we create new independent variables to account for all potential two and three-variable interactions. Note that these new variables undergo the described filtering process, i.e., those showing a correlation < 0.8 with E_{ad} or with a VIF > 10 are removed.

Finally, we eliminate constant or almost constant variables. These are variables with a variance of less than 0.01 since they won't help us estimate energy consumption better.

Table 2 shows the final list of independent variables we use as input to build our energy estimator model with their respective VIF factor. Figure 4 depicts the correlation between each pair of variables.

4.2.3 Model. We want to build a model that, based solely on an ad's parameters available upon rendering the ad, can predict how much energy will be consumed by rendering it. We trained our model using the 25k ads from our *automated* dataset. Our purpose is to obtain a sufficiently accurate model that can estimate the energy consumed by ads in nearly real time.

As discussed above, the *automated* dataset used to build our model includes ads which are not typically shown to users (due to the fact of our method involves the collection of ads by software and not by user browsers) and thus it may present some biases and not be fully representative of ads shown to actual users. To assess the generality of our model, we validate it with the 598 ads from our *human* dataset. We compute the nE_{ad} of each of the ads in our *automated* and *human* datasets in our lab experiment setup as described in Section 4.1.2.

Validation	Model	Linear	Random Forest	Gradient Boosting
Windows Desktop	R2	0,97	0,56	0,56
	RMSE	2,16	6,02	6,00
Ubuntu Desktop	R2	0,89	0,27	0,27
	RMSE	6,27	14,44	14,41
Ubuntu Laptop	R2	0,93	0,88	0,91
	RMSE	2,93	2,46	2,13
Window Laptop	R2	0,67	0,36	0,42
	RMSE	5,78	6,81	6,48
Mac Laptop	R2	0,89	0,62	0,66
	RMSE	2,92	3,82	3,66

Table 3. Validation with real ads.

After analyzing different models presented in Table 3, we conclude that the best estimator for nE_{ad} is a linear model. Table 3 displays the RMSE and R^2 scores for the different considered models. In this table, we present the results of comparing our model's predictions to actual readings from the 5 devices used in our validation setup. As shown in the table, our model performs sufficiently well regardless of which of the 5 devices it is compared against. Our estimator for determining the energy consumption of an online advertisement is effective, as evidenced by our Linear Regression model. One of the key advantages of our method is that we were able to develop a transversal estimator that works in a variety of relevant setups.

4.3 Limitations: What About the Mobile Ecosystem?

CodeCarbon does not work in mobile devices. This limitation may represent a concern. However, one of the essential goals of defining a normalized measure (nE_{ad}) is that it is related to the device's usage, taking into consideration just the increase in energy of the ad relative to the device's base

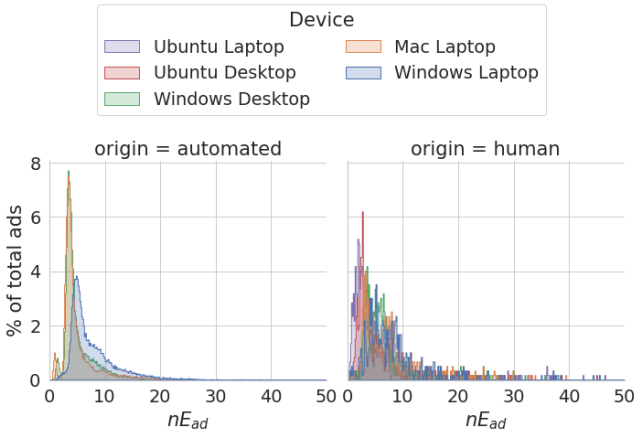

Figure 5. Distribution of energy consumption of ads by the device. On the left chart, the distribution of energy of ads gathered from automation. On the right, real ads gathered from real users.

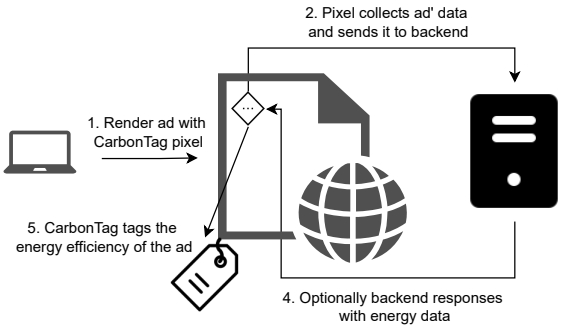

Figure 6. Visual representation of CarbonTagfunctionality.

consumption. As a result, this metric is applicable to mobile devices and any other hardware/OS combination not mentioned in this article. As future work, we plan on measuring the energy consumption of ads rendering in the mobile ecosystem, however owing to the complexity of measuring ad's energy consumption in a mobile device, we present a global method that can be extrapolated to any device in this paper.

This is the reason behind the normalized energy metric nE_{ad} based on the device+browser's base usage. Furthermore, another reason why there are two new devices (Windows Laptop and Mac Laptop) not included in the *automated* dataset, is in fact to verify our results against devices not included in training the energy estimation model. The goal is to demonstrate that the model provided in this study predicts unseen and real data from new hardware/software combinations in a realistic manner.

5 Ad Energy Labeling System

Making the ad industry and the general public aware of complex processes such as the energy consumption of the ad rendering process is difficult.

However, we have examples, which show us the path to generate this awareness. For instance, the EU's use of energy labeling framework [24] has proven that a simplified labeling system on the energy consumption of appliances creates the right awareness in the general public.

Following this successful use case, we propose a labeling system for ads. This labeling system (or an alternative one) could be used by the ad techindustry and users to filter out ads that consume an excessive amount of energy.

We use data from the energy consumption of the 25k ads from our *automated* dataset to develop a scale on which ads can be labeled as more or less energy efficient. According to Figure 5, the majority of the rendering of the advertisement is between 0-10 nE_{ad}. We propose to use a simple labeling scale system based on these distributions. We mimic the EU energy labeling framework and create 7 categories, from A (most energy efficient) to G (least energy efficient).

We propose the following labeling process according to the distributions shown in Figure 5. *A*: 0-1 nE_{ad}, *B*: 1-3 nE_{ad}, *C*: 3-6 nE_{ad}, *D*: 6-10 nE_{ad}, *E*: 10-15 nE_{ad}, *F*: 15-25 nE_{ad}, *G*: >25. nE_{ad}

6 CarbonTagsystem

CarbonTagis the system that we have implemented to accomplish our goal, i.e., measuring the energy consumption of the ad rendering process in the device in real time and at scale. In this section, we describe two potential practical implementations of CarbonTagreferred to as *CarbonTagServer-based* and *CarbonTagServeless*. In addition, we conduct the performance evaluation to assess whether it meets the requirements imposed by the current online advertising ecosystem. To conclude the section, we discuss its readiness and flexibility to be operational.

6.1 CarbonTagServer-based

This proposal represents the standard implementation of CarbonTag. CarbonTagServer-based is formed by two main components: the ad tagand the backend server. Figure 6 shows a visual representation of our system.

Ad tags are a standard technique used to insert code in ads or in the iFrame(s) containing the ad. These ad tagare typically JavaScript code that interacts with the browser to make different types of measurements and collect data, which are then reported to third-party servers.

In CarbonTag, the ad tagis a JavaScript code responsible for collecting the parameters (described in Section 4) to be fed to the model as independent variables. The parameters are gathered from native JavaScript libraries, mainly based on the Performance interface [25]. This library provides access

to performance-related information for the ad, using multiple APIs, such as Navigation Timing API, Resource Timing API, etc., which report the information about the loading time of each resource and its associated metrics.

The ad tagsends the collected parameters to the backend server, which upon the reception of the parameters runs the model and obtains an estimation of the energy consumption and the energy label of the specific ad being measured. These results are stored locally and eventually, the server can also send the results back to the ad tag(for instance) to show the energy label to the user.

6.2 Serverless CarbonTag

We have implemented an alternative version of CarbonTag, which does not use a backend server. In this version, the ad tagcollects the parameters for the model and also runs the model locally in the device to obtain the estimation of the energy consumption of the ad rendering process.

The serverless version of CarbonTagoffers few advantages over the standard version. On the one hand, it requires fewer compute resources, since it does not require the implementation of any sort of communication with a third-party, and the backend infrastructure of the system is not needed. On the other hand, it limits the amount of data shared by ads with third parties, which enhances privacy.

On the downside, the serverless version requires encoding the code of the model within the ad tag, thus not allowing to embed new heavier models that could appear in the future to estimate energy consumption. Also, the serverless version does not send ads' information to a backend, which could be useful for statistics and future analysis of ads performance or model retraining. Since the ad tagis by definition a light piece of code, the model encoded must be likewise light, limiting the type of models to be used to simple ones like linear regressors. The linear regression model can be implemented using around 10KB offering an R^2 between 0,67 and 0,97 (See Table 3). Instead, more complex models such as decision trees require at least a MB to be embedded in a pixel, which is not feasible.

6.3 Performance evaluation

The ultimate goal of CarbonTagis to measure ad energy consumption in real time and at scale. This requires meeting two specific requirements that the ad techecosystem would impose on our solution in order to be valid in an operational environment: real time operation and scalability. Finally, another important requirement is that CarbonTagshould be very light in terms of energy consumption.

6.3.1 Real time performance. To evaluate the real time requirement in the server-based version, we have measured the time elapsed since the instant the ad rendering process starts until an estimation of the energy consumed by the ad is received by the ad tag. This process considers the time required by the ad tagto collect the parameters, the communication with the backend server, the time required by the backend server to run the model, and the communication of the result to the ad tag. Similarly, we have also measured the time required by the serverless solution to obtain the energy consumption metric. We have emulated the described process in our lab setup for the *human* dataset.

Figure 7 shows the CDF of the time required by CarbonTagto perform the energy consumption evaluation both in the Server-based and Severless less versions. In particular, the figure shows the difference between the load time of the ad with and without CarbonTag, which captures the time overhead added by the use of CarbonTag. We observe that the median is 0.0 ms for both CarbonTagimplementations. Moroever, the 90-percentile is 0.23 ms and 0.08 ms for the server-based and the serverless version, respectively. Our results confirm that CarbonTagcan operate in real time adding a negiglible timing overhead over the current operation of ads rendering process.

6.3.2 Scalability. The server-based architecture of CarbonTagis a standard model used in ad tech(and the Web in general), which can easily scale with the number of requests. We have made a stress test in our lab to assess the expected number of queries that a commodity machine acting as CarbonTagbackend server may handle. In particular, we used a machine with 56 Cores, and 126 GB RAM, in a 10 Gbps connection at our lab. We clarify that the serverless solution runs locally and the requests scalability issue does not apply in this case.

We created a flow of ad tagrequests at different rates. We assessed that our commodity server was able to handle ad tagrequests rates generating up to 940 Mbits/sec of bandwidth. This result confirms CarbonTagoffers the required scalability to operate in the current programmatic advertising ecosystem.

6.3.3 Energy consumption. A solution meant to measure energy consumption, must guarantee to have a close to negligible energy footprint. Hence, CarbonTagshould present a low energy consumption.

To assess this, we have measured the energy consumption of the ads from our *human* dataset with and without CarbonTag. Figure 8 shows the distribution of the extra energy consumed by ads when running CarbonTagfor the different devices considered in our lab environment both in the server-based and serverless implementations. We conclude that the impact of CarbonTagis negligible in terms of energy consumption. Indeed, for a representative set of ads, the energy consumed by the ad including CarbonTagis lower than its equivalent without CarbonTag. This indicates that there are other factors, different from CarbonTag, which influence the energy consumption of an ad at a given instant (most likely linked to the OS and browser).

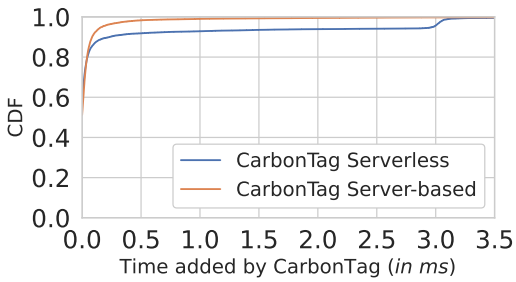

Figure 7. Cumulative Distribution Function of the time taken by CarbonTagto get the parameters and process the model, under two scenarios, online vs offline.

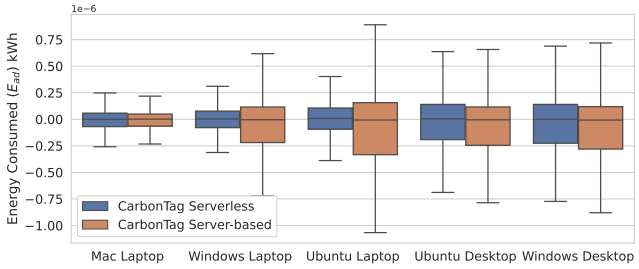

Figure 8. Performance of the solution CarbonTag, in both scenarios, loading the model online vs loading the model offline. Represents the consumption compared with the Ad.

6.4 Readiness and flexibility

CarbonTagis implemented using standard techniques from ad tech, i.e., and ad tagcommunicating with a backend server. Hence, any player from the online advertising ecosystem (advertisers, agencies, DSPs, ADXs, etc) can seamlessly integrate CarbonTagin their regular operation. In essence, they should add an extra ad tagto their operation.

In addition, CarbonTagis very flexible and adaptable. As in any machine-learning based system, the model needs to be retrained and eventually modified. If this occurred, there are two scenarios to be considered: 1) If no new parameters are used in the model, the only change would be to implement the new version of the model; 2) If the new version of the model uses a different set of parameters, then the ad tagneeds to be modified to collect the new set of parameters. In this case, we need to deploy a new version of the model and a new version of the ad tagneeds to be distributed across the companies using CarbonTag. As we explained earlier, any new version of the model will be valid to run in the backend server part of the server-based solution. However, depending on the complexity and size of the new model, it may not be feasible for the serverless solution.

7 Global Impact of Ad Energy consumption on the device

In Section 4, we have presented, to the best of our knowledge, the largest dataset used so far to measure the energy consumption of the ad rendering process of an ad. This is the energy that the ad delivery process uses in the device.

The primary goal of our lab measurements was to create a large-scale ground truth dataset to train our model. In this section, we take advantage of our effort to provide a ballpark estimation of the aggregated energy consumed by the ad rendering process of hundreds of billions of ads served every day.

According to our measurements, the median energy consumed by the ad rendering process of a single ad is 4e-6 kWh in the *automated* dataset and 3e-6 in the *human* dataset. More specifically, in the *human* dataset, the lowest consuming device, Windows Laptop, had a median energy consumption of 3e-7 kWh per ad, and the greatest consuming device, Windows Desktop a median of 8e-6 kWh per ad.

Moreover, despite there are no scientifically proven figures, industry reports indicate that each user may be delivered between 6,000 and 10,000 online ads per day [26]. Using these values as a reference and considering the devices reporting the lowest and greatest energy consumption of our dataset, the ads delivered to every single user would consume just in the rendering process between 1,8 and 80 e-3 kWh per day.

Furthermore, sources report that more than 5 billion people (or 63 percent of the global population) use the Internet [27]. If we consider this user base as a reference for our estimation, online ads would consume between 9e6 kWh and 400e6 kWh (9 million kWh to 400 million kWh) per day on user devices. As a result, online advertisements are expected to consume between 3e9 kWh (3 billion kWh) and 146e9 kWh (146 billion kWh) per year. To put these values in context, we can compare them with the overall energy consumption of different countries. If we do this exercise, we observe that the rendering of online advertisements consumes an amount of energy in the same range as countries like Luxembourg (6,5 Billion kWh), Portugal (49 billion KWh), or Sweden (131 billion KWh) [28].

As we have discussed above, it is expected that the device part (i.e., the ad rendering process) of the energy consumed by the ad delivery process is smaller than the correspondent network part, which involves several important factors. Under this assumption, we conjecture that the overall energy consumed by the delivery of ads might be in the order of tens to thousand of billions kWh per year.

8 Related Work

The increase in energy consumption brought on by the development of the ICT industry over the past few decades has to be investigated. The ICT industry currently accounts for

a sizeable portion of global energy consumption and greenhouse gas emissions, which has an impact on the environment. Numerous studies in the literature have examined this issue and highlighted the importance of reducing the amount of energy we use in favor of more environmentally friendly methods. Researchers concur that emissions and energy use from digital gadgets would likely rise in the near future. The ICT emissions share footprint will quadruple from the value of 2007 to 3-3.6% by 2020 and to 14% of the 2016 level by 2040, according to Belkhir et al. [29]. Additionally, Andrae et al. [30] mention predictions that CO2 emissions in the digital industry will increase between 2020 and 2030. They predict that between 82 and 96% of the rise in CO2 emissions in this decade would be attributed to computing and the digital industry.

Individual device usage is closely tied to the rise in this energy consumption. Ercan et al. [31] estimated the aggregate power consumption of digital devices like personal computers, telephones, communication networks, and data centers to be between 4 and 10% of the total consumption. According to Ruiz et al. research [32], end-user device emissions account for the majority of the carbon footprint of wireless ICT networks. In addition, Belkhir et al. [29] predicted that by 2020, smartphones would account for 11% of all ICT use, outpacing the contributions of desktops, laptops, and displays taken individually.

This increase in end-user carbon emissions emphasizes the need for developing tools to comprehend ICT energy consumption to support moving toward more energy-efficient industrial approaches. Monañana et al. [33] for example, focus on providing a tool to achieve a more efficient computation at the software level by logging the energy consumption of various algorithms and then choosing the more energy-efficient algorithm for satisfying the requirements of an application.

The use of personal devices is inextricably linked to the use of the Internet. Furthermore, as stated in Section 1, one of the primary revenue sources for the internet is the online advertising industry. As a result, measuring the energy consumed by online ads is critical. Despite the fact that the online advertising ecosystem and the presence of available ad spaces continue to grow, according to [8], there is very little literature on the subject.

Pärssinen et al. [9] acknowledge that measuring the energy footprint on the Internet is difficult due to the complexity of the network. Hence, calculating the energy consumption of online advertisements is not a simple task. To estimate the energy usage of any online service, they present a framework in their study as a result. They later estimate the energy usage of the online advertising ecosystem using this paradigm. They discovered that between 20 and 282 TWh of energy was consumed by online advertising in 2016. The total amount of energy used by all Internet-related mediums in that year ranged from 791 to 1334 TWh.

Similar to our proposal, two works measure the cost in terms of energy consumption of online advertising. Simons [10], simulating the actual usage of the web, measures the energy consumption with the measuring device Voltcraft Energy Monitor 3000. This device can measure with a resolution of 1W. We tried this approach and concluded that the results on the devices are not deterministic in the measurements, and the method is not scalable to multiple devices and environments.

There is another related paper from Prochkova et al. [11] where the authors measure energy consumption by comparing the power usage of mobile phone games with and without advertisements. Again, this paper uses a measuring device that is not comparable to our solution, which can measure each ad in real time.

Finally, Scope3 [34] uses a qualitative approach without the utilization of any direct measurement. That is using a number of constant variables of the energy consumption given by the different stakeholders like Google, Criteo, Pubmatic, etc. Hence, the accuracy of their estimation will depend on the accuracy of the used qualitative assumptions and the accuracy of these companies, whose methodology is not disclosed. Instead, our methodology measures the parameters associated with the ad rendering process on different devices and OSs. Moreover, while Scope3 provides energy consumption estimation at an aggregate level for different ad stakeholders, CarbonTagprovides energy consumption of the ad rendering process at the granularity of individual ads. We propose that advertisers and adtech companies will benefit from implementing both kinds of methods: those similar to Scope3 (top-down), and those similar to CarbonTag (bottom-up).

Putting all of this together, this paper presents an approach that measures the energy consumed by online advertisements, which run and consume energy from end-user devices, and provides an energy estimation ad tagto classify those advertisements that are less energy efficient, with the goal of limiting their display. To the best of the authors' knowledge, this is the first work that presents a methodology and a tool for estimating the energy consumption of on ads in real time at the individual ad level.

9 Conclusion

We investigated the energy consumption caused by online ads in this study. We investigated a method for calculating the energy online ads use when rendering, decoupling advertisement rendering from other computer operations. To analyze the energy used by an advertisement, software or hardware must be installed. However, as we have demonstrated, doing so for each and every advertisement would necessitate a significant shift in a well-established ecosystem such as the online advertising environment. It would also compel users to install new software and/or hardware, which

would require them to drastically alter their behavior. For these reasons, we decided to develop a methodology that transparently calculates the energy required by an ad using only the information available during the ad's rendering process, while not interfering with its current operation in any way.

As a result, we investigated the variables most closely related to energy consumption and created an estimating model that predicts the energy consumed by an online advertisement. We examined a sample of 25k advertisements and compared our findings to 598 online ads obtained from real people browsing the Internet. For this purpose, we used software to measure the energy used by advertisements across three different devices and operating systems. Then, without using any outside sources, we created our ad energy model, which calculates consumption solely based on the advertisement's internal rendering characteristics. Overall, our model achieves a great performance with an RMSE between 2,16 and 6,27 and a R^2 between 0,67 and 0,97 working across different combinations of operating systems and devices.

Finally, we created CarbonTag, a novel system leveraging a piece of code, an ad tag, that can be used during the rendering of any online ad to estimate the energy consumption of individual ads in real time and at scale. This would allow the ad tagto determine how energy efficient a given online ad is. Furthermore, we proposed a standardized meter for classified advertising on a scale of less to more effective (from A being ads that are the most energy efficient to G being the least), as well as the development of tools by intermediaries and ad techactors that would allow users to specify the types of ads with which they wanted to be exposed to based on their energy efficiency. This is an important aspect of our research because, according to our estimations, just the rendering part of online ads may consume the same amount of energy per year as a whole country like Luxembourg (if we consider the bottom range of our estimation) or Sweden (if we consider the top range). The network part of online ads will consume importantly more energy than the device part.

To the best of the authors' knowledge, this is the first piece of work that estimates the energy consumption associated with rendering online ads and proposes a standardization model that can be implemented to encourage the online advertising ecosystem and community to favor the display of more energy efficient ads.

References

[1] BP.

[2] U. E. I. A. (EIA), Jul 2022.

[3] M. Hübler and A. Löschel, "The eu decarbonisation roadmap 2050—what way to walk?," *Energy Policy*, vol. 55, pp. 190–207, 2013.

[4] J. Aslan, K. Mayers, J. G. Koomey, and C. France, "Electricity intensity of internet data transmission: Untangling the estimates," *Journal of Industrial Ecology*, vol. 22, no. 4, pp. 785–798, 2018.

[5] W. Carbon, "The original website carbon calculator."

[6] J. Kettle, "The internet consumes extraordinary amounts of energy.."

[7] N. Jones, "How to stop data centres from gobbling up the world's electricity."

[8] The Interactive Advertising Bureau (IAB), "Internet Advertising Revenue Report 2021," Apr. 2022. accessed on 19 July, 20222.

[9] M. Pärssinen, M. Kotila, R. Cuevas, A. Phansalkar, and J. Manner, "Environmental impact assessment of online advertising," *Environmental Impact Assessment Review*, vol. 73, pp. 177–200, 2018.

[10] R. Simons and A. Pras, "The hidden energy cost of web advertising," in *Proceedings of the 12th Twente Student Conference on Information Technology*, pp. 1–8, Citeseer, 2010.

[11] I. Prochkova, V. Singh, and J. K. Nurminen, "Energy cost of advertisements in mobile games on the android platform," in *2012 Sixth International Conference on Next Generation Mobile Applications, Services and Technologies*, pp. 147–152, IEEE, 2012.

[12] iab.TECH LAB, "Openrtb. version 2.6- released april 2022."

[13] JICWEBS, "Viewability product principles."

[14] MRC and IAB, "Mrc viewable ad impression measurement guidelines."

[15] A. Pastor, M. Pärssinen, P. Callejo, P. Vallina, R. Cuevas, Á. Cuevas, M. Kotila, and A. Azcorra, "Nameles: An intelligent system for real-time filtering of invalid ad traffic," in *The World Wide Web Conference*, pp. 1454–1464, 2019.

[16] M. Marciel, R. Cuevas, A. Banchs, R. González, S. Traverso, M. Ahmed, and A. Azcorra, "Understanding the detection of view fraud in video content portals," in *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[17] A. Noureddine, R. Rouvoy, and L. Seinturier, "A review of energy measurement approaches," *ACM SIGOPS Operating Systems Review*, vol. 47, no. 3, pp. 42–49, 2013.

[18] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, 2019.

[19] "Codecarbon.io."

[20] M. Inc., "Top 500 most popular websites in the world based on domain authority."

[21] C. Iordanou, N. Kourtellis, J. M. Carrascosa, C. Soriente, R. Cuevas, and N. Laoutaris, "Beyond content analysis: Detecting targeted ads via distributed counting," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pp. 110–122, 2019.

[22] Selenium, "Selenium automates browsers."

[23] E. Vittinghoff, D. V. Glidden, S. C. Shiboski, and C. E. McCulloch, "Regression methods in biostatistics: linear, logistic, survival, and repeated measures models," 2006.

[24] E. Union, "About the energy label and ecodesign."

[25] D. M. MDN, "Performance."

[26] S. Carr, "How many ads do we see a day in 2022?."

[27] D. Reportal, W. A. Social, and Hootsuite, "Digital 2022: Global overview report."

[28] U. E. I. A. (EIA), "International electricity consumption."

[29] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of cleaner production*, vol. 177, pp. 448–463, 2018.

[30] A. S. Andrae, "Hypotheses for primary energy use, electricity use and co2 emissions of global computing and its shares of the total between 2020 and 2030," *WSEAS Transactions on Power Systems*, vol. 15, no. 50-59, p. 4, 2020.

[31] M. Ercan, J. Malmodin, P. Bergmark, E. Kimfalk, and E. Nilsson, "Life cycle assessment of a smartphone," in *ICT for Sustainability 2016*, pp. 124–133, Atlantis Press, 2016.

[32] D. Ruiz, G. San Miguel, J. Rojo, J. Teriús-Padrón, E. Gaeta, M. Arredondo, J. Hernández, and J. Pérez, "Life cycle inventory and carbon footprint assessment of wireless ict networks for six demographic areas," *Resources, Conservation and Recycling*, vol. 176, p. 105951, 2022.

[33] J. M. Montañana Aliaga, A. Cheptsov, and A. Hervás, "Towards energy efficient computing based on the estimation of energy consumption," in *Sustained Simulation Performance 2019 and 2020*, pp. 21–33, Springer, 2021.
[34] Scope3, "Decarbonizing media and advertising."