



HAL
open science

The Impact of Green Feedback on Users' Software Usage

Adel Nouredine, Martín Diéguez Lodeiro, Noëlle Bru, Richard Chbeir

► **To cite this version:**

Adel Nouredine, Martín Diéguez Lodeiro, Noëlle Bru, Richard Chbeir. The Impact of Green Feedback on Users' Software Usage. *IEEE Transactions on Sustainable Computing*, 2023, 8 (2), pp.280-292. 10.1109/TSUSC.2022.3222631 . hal-03855088

HAL Id: hal-03855088

<https://hal.science/hal-03855088>

Submitted on 2 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Impact of Green Feedback on Users' Software Usage

Adel Noureddine, Martín Diéguez Lodeiro, Noëlle Bru, and Richard Chbeir

Abstract—The rise of the energy impact of software systems requires the need to optimize and reduce their energy consumption. One area often neglected is the important role played by users to drive energy reductions. In this paper, we aim to reduce the energy impact of software by pushing end users to change their software usage behavior, through raising awareness and providing software green feedback. We present a comprehensive and detailed field study of the impact of green feedback on software usage by end users, and the efficiency of green feedback on software behavioral change, using a distributed architecture aimed at providing accurate green feedback in real time. We find that green feedback helps in raising awareness about software energy, and on the willingness of users to apply energy-efficient changes. However, we also find that users lack the knowledge and tools to properly adopt lasting and energy-effective behavioral changes.

Index Terms—Energy Consumption, Behavioral Change, Green Software, Field Experiment



1 INTRODUCTION

With the widespread usage of smart devices, it is expected that 500 billion devices will be connected to the internet by 2040 [Evans(2011)]. With this growth, the greenhouse gas emissions (GHGE) of Information and Communication Technologies (ICT) is expected to grow to up to 14% by 2040, from their 2% levels in 2007 [Belkhir and Elmeligi(2018)]. In the European Union, ICT are responsible for 4% of CO2 emission, and up to 10% of electricity consumption in 2015 [ICTFOOTPRINT.eu(2015)].

Therefore, the need to optimize the energy consumption of software and devices, along with pushing users to reduce the usage of ICT devices, is crucial for a sustainable future. Today, most energy management approaches in computing, whether it is software systems, data centers, or cyber-physical environments, focus on technological advancements or optimizations. However, very few approaches integrate end users into the energy optimization equation, and most of these approaches target energy consumption in buildings or smart homes, where energy reductions are achieved in non-computing equipment (such as refrigerators, lightning, HVAC systems, etc.).

We argue that the next major shift in energy reductions needs to involve end users, for instance in relation with reducing the usage of ICT or changing users' behavior regarding ICT devices. In particular, changing the behavior of end users in using smart devices, and in particular software running on these devices, might lead to important energy savings. We decide to focus on software, such as applications or operating systems, which are easier to update than hardware devices. Also, any improvement to the

source code of software can scale well and be deployed to billions of devices. In addition, end users essentially interact with software, and the *smart* part of smart devices is their software.

In this paper, we aim to understand the impact of green feedback on users' software behavior. In particular, we want to study and understand, through a field experiment:

- What do users know about energy consumption of software?
- What perceptions do users have on software and energy?
- Will green feedback of software usage lead to behavioral change and energy reductions?

Many other factors impact energy consumption and/or users behavior. For instance, green feedback might raise awareness for users and progressively change their behavior over time. Users also may lack sufficient knowledge on what to do when energy consumption is high, besides turning off devices or stop using them. In our study, we decide to limit our investigation on two areas:

- Study the short-term behavioral change of end users after providing them with green feedback.
- Study the perception and user awareness in regard to energy consumption of software.

Our research question is, therefore: **can raising user energy awareness, through live green feedback, drive behavioral change in software energy consumption?** In particular, we want to understand if live green feedback can drive short-term behavioral change, and what type of change users think are appropriate.

Feedback is any information send by software systems to end users, related to the state of software. Energy feedback is specific feedback related to the energy or power consumption of software. An existing definition in [Karlin et al.(2014)] explains the difficulties of proposing a clear operational definition, and defines energy feedback as *information about actual energy use that is collected in some way and*

-
- A. Noureddine and R. Chbeir are with Université de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, EA 3000
Email: adel.noureddine@univ-pau.fr and richard.chbeir@univ-pau.fr
 - M. Diéguez Lodeiro is with University of Angers
Email: martin.dieguezlodeiro@univ-angers.fr
 - N. Bru is with Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LMAP
Email: noelle.bru@univ-pau.fr

provided back to the energy consumer. We extend this definition to cover other metrics and information than those of energy and power, specifically ecological and sustainability metrics (such as CO2 emissions, renewable energy usage, etc.).

In this paper, we conduct a field experiment on the impact of green feedback on software usage. We build a distributed architecture to provide live green feedback for end users, using web services to connect to electricity providers to calculate and provide up-to-date and accurate feedback on the power consumption of software and devices, and the price of this consumption and its CO2 emissions.

The field study consists of a power-monitored workload, along with a survey for end users. Our field study provides promising results on the role of green feedback, but sheds light on the challenges related to software behavioral change and the limitations of current tools and feedback visuals. To our knowledge, we believe our field experiment, conducted with a control group, is novel in the green computing and software engineering communities.

It is important to stress that our goal in this study is not to provide guidance or alternative software for users, nor do we aim to tell users what is the right course of actions to reduce energy, nor the correct behavior to adopt. Our study is an observation field experiment where we aim to observe, analyze and understand what are the factors impacting behavioral changes around software energy, and if green feedback can trigger a behavioral change in users.

Although we have ideas of what can impact software energy and user behaviors, and what users might do to reduce their energy footprint, we have no existing scientific data to back any assumption about end users' behaviors or to provide suggested recommendations for users in this study. We aim to collect and build solid scientific data to validate our research question and hypothesis, and that can be used by the scientific community for future studies.

The remainder of this paper is organized as follows: in Section 2, we explore existing approaches around energy-aware behavioral change. Next, we detail our field experiment in Section 3, describing our methodology in both the experimental setup and the survey. In Section 4, we discuss and analyze the results of our field study and provide recommendations on green feedback for software behavioral change. Finally, we conclude in section 8.

2 RELATED WORKS

Driving behavioral change in software and smart environments comprises many diverse challenges and understandings: from the inherent challenges of smart systems, to the ambiguous nature of software and its energy impact, and the challenging nature of human behaviors and understanding the factors that guide it.

Researchers in these three domains have tried to tackle energy consumption through various approaches. We outline the main related works in this section, and describe how our approach aims to address these challenges across software, devices and behavioral change.

In software, energy efficiency has been a rising concern for software developers [Pinto and Castor(2017)], architects [Bashroush et al.(2016)], and practitioners [Manotas et al.(2016)]. Existing approaches to manage the energy

consumption of software often involve source code optimization or refactoring, including optimizing data input and storage, memory usage, or network access [Ardito et al.(2015)], [Jagroep(2017)], [Zhang et al.(2022)]. Beyond the individual programs, software systems energy optimizations often target the entire computing architecture, from middleware [Noureddine et al.(2013b)], to servers and equipment in data centers [Level(2015)], or optimizing the workload in servers, virtual machines, and in distributed environments [Orgerie et al.(2014)], [Colmant et al.(2018)]. In addition, software approaches have been proposed to monitor the energy consumption of software [Noureddine et al.(2013a)] allowing developers key insights on which hardware component and which application is consuming energy. Other approaches allow deep energy investigation of software, providing energy consumption per source line of code, functions, or software modules [Noureddine et al.(2015)]. However, most of the existing approaches and solutions target computing experts, from software developers to system administrators, or automated systems and tools. End users are rarely involved in energy optimization strategies, and therefore little to no feedback is provided to users about their energy consumption. This observation holds true for IoT and embedded environments, where low-power and low-battery equipment are often managed by predefined algorithms or automated machine learning tools. End users are even less involved in reducing the energy consumption of smart connected devices, such as smart TVs, smart speakers, or other IoT devices, such as alarms or home security cameras.

In contrast, the literature is rich with approaches involving end users, and field studies on users behavior regarding energy consumption in buildings or smart homes. In a recent study, the authors reviewed the state-of-the-art of methods and approaches to energy efficiency in buildings [Paone and Bacher(2018)], and concluded that most solutions to influence occupants' behavior in buildings revolve around three main approaches: eco-feedback, social interactions, and gamification:

Eco-feedback focuses on providing occupants with relevant information about their energy consumption, whether this data is live or covers the occupants past consumption. The form of this provided feedback varies from direct energy consumption metrics, to prices or CO2 emissions, or an abstraction of energy consumption with the use of metaphors, such as the number of trees saved [Wood and Newborough(2007)]. A large-scale study involving 2000 households showed saving of up to 15% with eco-feedback [Vassileva et al.(2012)]. However, researchers warn of the limited effects of eco-feedback on long-term energy saving and behavioral change, as rebound effects might happen [Buchanan et al.(2015)], and suggest employing normative feedback for positive behavioral change in the long run [Anderson and Lee(2016)].

Social interactions allow users to compare their energy behavior and savings with other users in their household, building or neighborhood, or even online through dedicated platforms or social networks. Such interactions and social implications can lead to important energy savings of up to 55% [Jain et al.(2013)].

Finally, *gamification* is seen as a new approach to motivate

users to change behavior. It involves integrating users into social or interactive gaming experiences, where the emotional engagement of users helps push for new behaviors. Example of such games might involve suggesting specific actions for energy reductions, correcting bad energy usage, or motivations for tangible and social actions [Reeves et al.(2015)].

Fischer [Fischer(2008)] reviewed 26 projects in 10 different countries (USA, Japan, and 8 European countries), on the impact of feedback on household electricity. The study found that feedback should be useful for end users in order to achieve electricity conservation. This might be through: actual energy consumption, be given frequently, involves interaction with users and providing them with choices, break down consumption to specific appliances and equipment, be given for a long period of time, involve historical or normative comparisons, and be presented in an easy to understand and appealing way. However, the author warns that these recommendations might fail for certain demographics, such as heavy consuming users, or elderly people with lower technical skills and interests. Darby [Darby(2001)] studied feedback approaches in relation to household electricity, and found three main groups of feedback: 1) direct feedback such as using smart meters or displaying energy data on a display, 2) indirect feedback, such as using historical data or normative feedback in electricity bills, and 3) inadvertent feedback where knowledge is learned by association. More recently, smart devices and equipment are more frequently used to collect data in cyber-physical systems and environments, and assist in providing green feedback to end users. For instance, IoT devices were used in smart city trials with an aim to building greener cities [Sánchez et al.(2014)]. Overall, these approaches in regards to behavioral change follow the same main techniques of building, such as using eco-feedback or gamification [Lu(2018)], [Konstantakopoulos et al.(2019)], [Francisco and Taylor(2019)], [Orland et al.(2014)].

3 FIELD EXPERIMENT

In this section, we describe the methodology we applied in the experiments and user surveys. We want to understand whether raising awareness through live green feedback of the users' energy consumption will help achieve short-term behavioral change, and whether users' general knowledge of software is sufficient to help them make relevant behavioral changes, if any.

As we aim to study the impact of green visual feedback on user behavior, we set up a field experiment with two user groups. Our field study consists of two steps: first, users participate in a common experiment where their energy consumption is monitored, then, they fill up a survey to collect their feedback and knowledge on the experiment and green computing. In the next sections, we detail the methodology of our field study.

3.1 Participants

In June and July 2020, around a hundred students (95) from computer science and engineering degrees in Lebanon, participated in the field study and survey. They participated

in an optional course in their degree on the topic of Research Methodologies (explaining how research is conducted, from studying the state of the art to experimental studies). The lecturer explained how user-centric experiments are conducted and explained the goal of our experiment. Students had the choice in participation, where many accepted but some did not.

The field study was conducted for each student group during one class session (2 hours), with similar protocols for both control group and treatment group (where only the software collecting the metrics was different). All students were tasked for a similar workload, and every student participated in the experiment during one class session. In total, 5 class sessions were needed for the experiment, with 2 classes being the control group (A1), and 3 classes for studying the impact of green visual feedback (A2), with overall a similar number of students in groups A1 and A2. Each class was either part of the control or treatment group (we did not divide a class) in order to avoid students influencing each other during the experiment and voiding the validity of our control group.

Participants were split in two for experiment A1 and A2, with 47 participants in group A1 answering the survey, and 48 for group A2. The majority of participants were males (74.47% for the control group A1, and 83.33% for A2). The average age is 22 years old (youngest at 21 and oldest at 27 years). Participants were students from computer science and engineering degree, with a minority having an additional part or full-time job (18.9%). The majority of participants, although studying in computing fields, have one or two mobile devices: 33.70% have 1 device, 36.96% have 2 devices, 15.22% have 3 devices, while only 14.13% had four or more devices. Overall, participants are interested in ecology, with a majority of 71.58% choosing the answer that "ecology is a major concern for our planet (pollution and global warming)", with 6.32% saying they are only interested in pollution but not global warming, and 22.11% saying they are not interested in ecology. Finally, a majority of participants (75.53%) think that radical (43.62%) or gradual (31.91%) actions need to be implemented to reduce energy consumption. 15.96% answer that some actions are needed to maintain energy consumption at current levels, while 8.51% answer that no change is needed as the status quo is fine. Overall, there are no significant differences between participants in the control group A1 and the experiment group A2.

3.2 Workload

Every participant had to perform a similar task in order to have a similar workload for comparison. Participants were asked to write a research document with heavy usage of online resources. Participants had a specific scientific topic, for example compression algorithms in IoT, and had to write a state-of-art document of 3 to 4 pages about it, within an hour and a half of class session. The produced document includes an introduction, the description of the problem, the existing works and solutions along with a comparative study, and a conclusion.

Participants used two sets of software: cloud-based tools, such as the online latex editor OverLeaf, and web browsers

to search for research papers, documents, multimedia content, etc., and on-device tools, such as word processing software, PDF or image viewers, etc. Participants had the liberty to choose the software they wish, such as using Google Chrome or Mozilla Firefox for the web browser.

3.3 Experimental Setup for power monitoring and feedback

All participants used a mobile device (a laptop) running mostly Microsoft Windows, with a few using a Macintosh device running macOS.

We built a power monitoring software, along with a distributed architecture, that collects the following metrics, every second:

- **CPU power consumption** of the user’s device, in watts. This metric is obtained using Intel Power Gadget’s APIs, and therefore participants had to install this software too.
- **Name and path of software** used by participants. Using Windows and macOS APIs, we collect the name of the software being actively used by users (the active window).
- **Title of the active window.** This also uses Windows APIs, and helps in understanding which website or tab the user is actually using. Therefore, it gives us insights on the type of activity executed by participants (e.g., writing Latex document, watching a video resource, etc.), as these can have an impact on energy consumption.
- **CO2 emissions** in grams, which is based on the global average CO2 emission of the electricity production in the country of participants, and the actual power consumption of the device.
- **Cost of electricity** in cents of euros, which is also based on the average cost of electricity in the country of participants, and the actual power consumption of the device.

Our software uses a mashup architecture which orchestrates different web services capable of collecting and obtaining: the current price of electricity, and the current CO2 emissions due to the generation of electricity. Our approach first identifies the geographic location of end users, and according to its region, obtains the relevant metrics. For instance, for a user located in France, our architecture would connect with the electricity providers (for instance, EDF) to get the latest real-time prices and CO2 emissions. Therefore, two users in different countries consuming the same amount of power, would be provided different green feedback for the price and CO2 categories. The approach could also integrate specific user electricity contracts and prices, such as peak hour prices, zero-carbon, enterprise or real-time priced contracts. For the purpose of the experiment, CO2 and electricity data are collected once from the electricity providers, and used throughout the experiment in order to minimize the impact of network on energy consumption and mitigate slow network connections for certain participants.

Current CO2 emissions and electricity price for the user workload is calculated by multiplying the current real-time power consumption (in watts) by the user location’s CO2 emissions (in grams per kWh), or by the price in cents per kWh, and we then adjust the unit metrics.

In addition, we developed a visual feedback tool which showed the power consumption, CO2 emissions and the

cost of electricity in real time in a small window. This window, shown in Figure 1, cannot be minimized or resized by participants, and stays on top of all other programs at the bottom right of the screen. In addition to textual values of the metrics, a graph is also shown which monitors in real time the power consumption of the device.

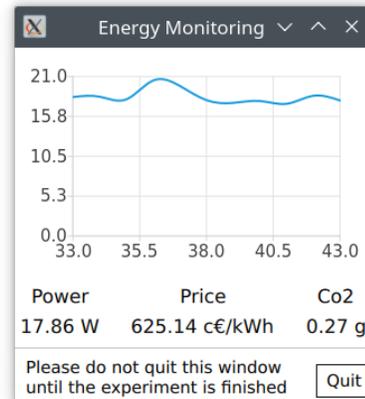


Fig. 1: The green visual feedback tool used in our experiments

End users will only see the green feedback provided in the tool (e.g., ecological and power metrics, and the evolving power chart). They will not know the power consumption of specific software, nor, for example, which one is causing a power spike. We designed our feedback to limit the provided information as our aim in the study is also to understand what users know about software energy and their perception of energy-consuming software.

Participants in the control group (group A1) were asked to install the power monitoring software and its dependencies, and run it in the background. This software will silently monitor power consumption of the device and collect software usage. Participants in the experiment group (group A2), were asked to install both software: the power monitoring one (the same as the control group A1), and additionally our green visual feedback tool which will display green feedback in a persistent window on the desktop (cf. Figure 1).

As lockdown for universities was still in place during the experiments, participants were asked to send the generated metric data (one CSV file and one TXT file) by email. However, due to technical problems installing the tools, or running and generating the data, and with issues with mapping the collected data to the anonymous survey, we ended up with 52 valid distinct power data files divided into 26 for group A1 and 26 for group A2.

We did not scientifically measure the overhead of either our power monitoring tool or our green feedback tool. The former is a simple two C++ program with one class and linking one external library (the Intel Power Gadget header), while the latter only adds a Qt GUI that displays a chart and 3 numerical values. We consider the overhead quite low and negligible, as our tools use existing energy monitoring approaches and APIs that has been shown to have low overhead in the literature [Noureddine et al.(2012)], [Fieni et al.(2020)].

In contrast, the distributed nature of our architecture might have an overall energy impact, as connecting to Internet APIs to get electricity prices and CO2 emissions might consume energy in the network equipment and servers. However, these remote APIs are not accessed continuously (ideally, once a day). We did not analyze the impact of the general architecture on energy consumption, as our focus are on end-user machines and on participants' behavioral changes.

Participants' Hardware and Software Profiles

All participants who send valid monitoring data used a Windows 10 laptop with 4 exceptions: three participants with Windows 8.1, and one with Windows 7, all of which were 64 bit devices. A few users had macOS laptops but none provided valid power data for the experiment.

All laptops were using Intel Core processors: two i3 models, nine i5 models, and the remaining were i7 models. The thermal design power (TDP) of these processors varied from the lowest at 15 watts (for 42 laptops), to up to 95 watts for one laptop using an i7-6700K CPU at 4 GHz, with 18 laptops with a TDP at 45 watts.

In terms of applications, and as the workload is document-oriented, participants mostly used a Internet browser, and office software. In particular, the majority of participants used Google Chrome with a few participants using Mozilla Firefox, Microsoft Edge and Opera Browser. The usage of the browser was extensive as participants use it for online search, email and messaging, viewing PDF files, and using the online Latex editor Overleaf. The majority also used Microsoft Teams for the class.

Other applications were used occasionally for specific tasks, such as Microsoft Office software (in particular, Word, Excel and PowerPoint), Windows Notepad for note taking, Adobe Acrobat Reader to view PDFs, Windows Mail for emails, WhatsApp Desktop for messaging between some participants, and the Windows Snipping Tool to take screenshots for the assignment.

3.4 Survey to collect user feedback

At the end of each experiment, participants were asked to fill in an anonymous online survey. In addition to the power data filename (which we collected to map the survey results to the power consumption data), participants had to answer a set of question regarding sustainability, green computing, and the experiment they are participating in.

We built our survey questionnaire with 4 main blocs around the following themes: behavior (*what we do*), opinion (*what we think*), motives (*why we do*), and identity (gender, etc.). We first asked identity questions, then factual questions, and finally motivation and opinion questions. We privileged closed questions as they are easier to answer by participants and to process by us, and to avoid ambiguities. We also avoided tendentious and emotionally charged questions in order to keep the questionnaire neutral. We built our multiple-choice questions to be as precise as possible by asking participants to classify their answers. In order to appreciate an impact of an answer and force a positioning, we built our questionnaire using a Likert-type scale without a midpoint (*i.e.*, without the **Neither agree nor disagree** answer).

We first asked participants questions about their identity: gender, age, education, and work status. These questions aim to understand whether age, gender or education levels have an impact on their opinion or behavior regarding energy consumption and sustainability in general. We then asked participants on how many mobile devices they own, and their interests in ecology and their view on global warming and if any action needs to be done for energy reductions.

Participants were then asked a few questions on the experiment they participated in. Participants in groups A1 and A2, answered questions on their perception of the energy cost of their experiment. They also had to rate the energy costs (from very low to very high) on a set of software categories (office programs, internet browsers, scientific or programming software, and the operating system itself), and then rank the most consuming categories and individual software.

Participants in group A2, had additional questions about the green visual feedback. In particular, we want to know if the visual tool was helpful or distracting, which form of feedback was most helpful (visual graph or textual numbers), and which metric was most useful (power consumption, CO2 or price). We then asked participants if they changed their software behavior based on the visual tool feedback, and if so, how they changed behavior.

Overall, for a participant, the process to participate in our field study is as follows:

- 1) The lecturer explains the experiment and its goals,
- 2) Students accept to participate or not,
- 3) Those who accept are provided with a written guide about the experiment and how to install the relevant monitoring software,
- 4) Participants in the control group install the monitoring software, while those in the treatment group install both monitoring software and the visual feedback tool,
- 5) After the experiment ends, all participants answer the survey questionnaire (with additional questions for the treatment group),
- 6) Finally, participants send back all data files from the monitoring software by email.

4 RESULTS AND ANALYSIS

In this section, we outline the results of our experiments, including both the energy monitoring data and the survey answers.

4.1 Perception of energy consumption

This part of the survey was answered by participants from both group A1 and group A2. We first asked our participants to rate the energy consumption of their experiment. To the question *"How do you rate the energy consumption of the session you just participated in?"*, a majority of participants rated their session low (49.47%) or very low (11.58%), with 36.84% rating it as high, and only 2.11% rating it as very high. However, these numbers vary between the control group (A1) and the experiment group (A2), as seen in Table 1. Participants who had our live green feedback window, rated

their energy session higher with 50% answering *high*, in comparison to only 27.66% in group A1 answering high or very high. In contrast, none in group A2 answered very high, and a bigger percentage (14.58%) stated very low, in comparison to 8.51% giving the same answer in the control group A1. Statistically, the difference in answers between groups A1 and A2, is significant. This shows that participants have a different point of view of their power consumption when shown live green feedback.

This initial finding suggests that users, even computer savvy ones such as computer science and engineering students, have a difficulty in judging how much their devices and software actually consume in terms of energy. We observe that our participants tend to minimize their power consumption (nearly three quarters in the control group), unless shown their actual power consumption (such as our live green feedback tool). In this latter case, the perception of participants shifted massively into the high consumption, and, to a lower extent, to very low. For instance, users of the control group A1 with a laptop having a CPU with a TDP (Thermal Design Power) of 15 watts, massively rated their session as low (77.78%), while participants with the same TDP in group A2 were 42.85% to answer high (cf. Table 2). The numbers are even clearer for higher TDPs. For instance, for participants at 45 watts TDP, 80% of group A1 rated their session low or very low, compared to 77.78% rating it high in group A2.

We then asked participants to rate their perception of the energy consumption of four main software categories: office programs, internet browsers, scientific or programming software, and the operating system. Participants in both groups A1 and A2 has similar answers. Overall, the operating system had the most very high ratings (26.09%), and an important majority (77.17%) rating it high or very high (cf. Table 3). These numbers are similar for scientific or programming software (typically, IDEs, compilers, Latex, Matlab, etc.), with 75.79% rating it high or very high. Surprisingly, internet browsers were also rated high, with 66.32% rating it high or very high, albeit a lower number rated it very high (10.53%). In contrast, office programs were the ones least seen as high energy consumers, with 38.79% rating them as high or very high. Even better, office programs had the highest percentage of participants rating them very low (12.63%), while no one rated the operating system very low.

These numbers might suggest that users, regardless of having green feedback, have some perception of which software might consume more. Office programs are seen less consuming as they mostly use textual input, with limited graphs or figures. In contrast, internet browsers today handle rich and complex multimedia websites, and are known to use higher amount of RAM memory and CPU cycles compared to word processing programs. Scientific and programming software, for our computer and engineers participants, can also be perceived as sluggish and memory or CPU-intensive. However, the surprising finding is that the majority of participants consider the operating system as the one having a high or very high energy consumption.

In order to investigate more this perception, participants were asked to classify the most energy-consuming software from a list of proposed ones (cf. Table 4): internet browser,

word processor, presentation program, email client, file or pdf viewer, IDE, and communication software. Most participants rated three software in the first spot: IDE (Eclipse, Visual Studio, etc.) at 33.33% of participants, Communication (Skype, Teams, Discord, etc.) at 28.89%, and Browser (Chrome, Firefox, Edge, etc.) at 25.56%. The other software didn't pass 7% and no one rated email clients (Outlook, Thunderbird, etc.) at the first spot.

The second spot was also taken by IDE (29.47%) and Communication (28.42%). However, only 5.26% placed Browser, while word processor was ranked second for 18.92% of participants. The third place was taken by Browser (30.53%) and, surprisingly, file or PDF viewer at 20%, with presentation program at 16.84%. The next two spots were shared between word processor and presentation program (21.62% then 29.73% for the former, and 21.05% then 22.11% for the latter). Finally, most people ranked file or PDF viewer as the least consuming with 35.79% ranking it last.

Finally, most participants consider cloud computing or data centers the most consuming part of the computing industry at 62.11%, with network equipment second at 12.84%, desktop computers third at 12.63%, and finally mobile computing last with 8.42%. This shows that computer-savvy participants are aware of the energy weight of cloud infrastructure (data centers and networking). However, it also shows that the energy consumption of mobile and IoT devices are still being considered individually, therefore, participants have a perception that these devices have a low impact on global energy consumption.

4.2 Perception of green feedback

Participants in the experiment group A2, had to answer an additional set of questions in the survey, in particular about their perception of the green feedback tool and their usage behavior of software and their devices.

We first asked participants what is their opinion and perception of the visual energy feedback (cf. Table 5). A majority of participants agree (58.33%) or strongly agree (33.33%) that the visual feedback was helpful to know when energy consumption is spiking. A similar majority agrees (59.57%) or strongly agree (14.89%) that energy consumption was higher than they thought, while a quarter (25.53%) disagreed. When we asked the opposite question (if energy consumption was lower than what they thought), 39.58% of participants agreed and 15.58% strongly agreed, while 35.42% disagreed and 10.42% strongly disagreed. These numbers show that, overall, participants thought they consumed more or less than their initial perception, therefore showing the importance of feedback in raising awareness. In contrast, around half of participants (47.92%) did not understand what each metric relates to. Finally, a majority of participants saw the window of our green feedback tool as distracting (57.78%).

We then asked users which energy metric in our feedback tool was the most helpful (cf. Table 6). 41.67% of participants answered that the power consumption metric in watts was the most helpful in understanding the green impact of their software and devices. Around a fifth (22.92%) answered that all 3 metrics were helpful (power,

Group	Very low or negligible	Low	High	Very High
Control group A1	8.51%	63.83%	23.40%	4.26%
Experiment group A2	14.58%	35.42%	50%	-

TABLE 1: Answers, per group, to the question: "How do you rate the energy consumption of the session you just participated in?"

Group/TDP	Very low or negligible	Low	High	Very High
Control group A1				
15 watts	5.555%	77.78%	11.11%	5.555%
17, 28, 35 watts	-	100%	-	-
≥ 45 watts	50%	33.33%	16.67%	-
Experiment group A2				
15 watts	19.04 %	38.09%	42.85%	-
≥ 45 watts	14.28%	14.28%	71.42%	-

TABLE 2: Answers, per group and per CPU TDP, to the question: "How do you rate the energy consumption of the session you just participated in?"

Software category	Very low	Low	High	Very High
Office programs	12.63%	51.58%	32.63%	3.16%
Internet browsers	4.21%	29.47%	55.79%	10.53%
Scientific or programming software	4.21%	20%	50.53%	25.26%
The operating system (Windows, macOS)	0%	22.82%	51.09%	26.09%

TABLE 3: Answers, per software category, to the question: "How do you rate the energy consumption of the following category of software?"

Ranking	Software 1	Software 2	Software 3
1st	IDE (33.33%)	Communication (28.89%)	Browser (25.56%)
2nd	IDE (29.47%)	Communication (28.42%)	Word processor (18.92%)
3rd	Browser (30.53%)	IDE (20%)	Presentation program (16.84%)
4th	Email client (43.2%)	Presentation program (18.95%)	Browser (11.58%)
5th	Word processor (21.62%)	Presentation program (21.05%)	File or PDF viewer (15.79%)
6th	Word processor (29.73%)	Presentation program (22.11%)	IDE (14.74%)
7th	File or PDF viewer (35.79%)	IDE (11.58%)	Word processor (10.81%)

TABLE 4: Answers to the question: "Rank the energy consumption of the following software"

price and CO2 emissions), of which a majority (42.86%) stated that power was more helpful than the other two. The price was selected for 16.67%, and 12.5% of participants didn't find any of these metrics helpful. Surprisingly, only 6.25% of participants found CO2 emissions metric helpful.

These numbers indicate that participants better associate power metrics to energy-related issues. Other metrics might add a layer of reasoning and comparison, therefore might render them less useful.

In contrast, participants were more attentive to the visual chart rather than the numerical metrics (cf. Table 7). 25.58% of participants considered the visual chart of our tool more helpful, in comparison to 16.28% for the metrics, while a majority of 58.14% answered both. Of the latter, when asked

to choose one of the two, a majority of 59.46% preferred the visual chart over the metrics (40.54%).

Participants also overwhelmingly agreed or strongly agreed (91.67%) that the power consumption graphs and metrics correlate to what they think their energy consumption is. However, this agreement does not translate with behavioral change in using software. When asked "How often did you change your software usage?", a majority of participants (62.5%) answered that they did not change their software usage. In addition, a fifth (20.83%) only changed their software usage when they were doing less important tasks (in relation to the workload). Only 16.67% of participants applied some sort of behavior change when they saw a power spike in our visual feedback tool.

However, when participants changed their behavior on every spike or temporarily, the applied actions were limited, mostly to a lack of knowledge of what to do (cf. Table 8). 54.17% answered that they switched software to reduce energy consumption, while 62.5% decided to reduce their usage time of some programs. In addition, 58.33% stated that they wanted to reduce their energy consumption but didn't know how.

These numbers indicate that participants know, essentially, only two ways to reducing their energy consumption: change software or reducing usage time. Both of these dimensions, tooling and temporal, imply that users must be armed with the knowledge of which software to go to, and for how long should a temporal reduction be applied.

In the next section, we discuss our findings and propose recommendations for a more effective behavioral change when using software.

Question	Strongly disagree	Disagree	Agree	Strongly agree
Helpful to know when energy consumption is spiking	-	8.33%	58.33%	33.33%
Energy consumption was higher than I thought	-	25.53%	59.57%	14.89%
Energy consumption was lower than I thought	10.42%	35.42%	39.58%	14.58%
The window was distracting	11.11%	31.11%	57.78%	-
I did not understand what each metric relates to	16.67%	35.42%	35.42%	13%

TABLE 5: Answers of participants in group A2 on how they rate the visual energy feedback.

Answer	Percentage
Power consumption in watts	41.67%
CO2 emissions	6.25%
Price in euros	16.67%
All of the above	22.92% (of which 42.86% for power, 28.57% for price, and 28.57% for CO2)
None	12.5%

TABLE 6: Answers of participants in group A2 to the question *Which energy metric was the most helpful?*

Answer	Percentage
Both	58.14% (of which, 59.46% for the chart, and 40.54% for the metrics)
The visual chart	25.58%
The numbered metrics	16.28%

TABLE 7: Answers of participants in group A2 to the question *Which type of feedback was the most helpful?*

	Strongly disagree	Disagree	Agree	Strongly agree
I switched software to reduce my energy consumption	2.08%	43.75%	39.58%	14.58%
I reduced the usage time of some programs to reduce energy consumption	2.08%	35.42%	54.17%	8.33%
I wanted to reduce my energy consumption, but I did not know how	6.25%	35.42%	43.75%	14.58%

TABLE 8: Answers of participants in group A2 to statements on software usage changes

5 DISCUSSIONS

5.1 Perception of energy consumption

Software power consumption in mobile devices vary depending on the device itself. A mobile phone running a cloud-based Latex editor will not have the same power consumption as a Raspberry Pi or a laptop, even for the exact same workload. As users tend to use multiple devices (two thirds of participants state having 2 or more mobile devices), we argue that raw metrics may not be the best approach to raise awareness or drive behavioral change. As applied in buildings [Wood and Newborough(2007)], metaphors may provide better understanding for users about their energy consumption. Such metaphors might be metric-based, such as trees saved or nuclear plants required for electricity production, or might be traffic-light style, such as showing a red symbol or phrase if energy consumption is deemed high. The latter classification of what is high or low energy consumption, is a challenging task for the software

research community. Some suggestions may include having a database of average energy consumption per workload, or per category of software, and can be used by an intelligent tool to show warnings to users based on their consumption and the importance of the workload.

From the results, there is a clear majority of participants ranking IDEs as the most consuming software. This finding may be biased to the fact that participants were computer science and engineering students using IDEs very often on not optimal laptop devices. Their perception of energy consumption might be influenced by the perception of slowness of IDEs, their rich feature set, or compilation time of programs. It would be interesting for the research community to extend our study to further understand this perception and what software developers can do to improve it, or to optimize their IDEs. In contrast, communication software was seen even more consuming than internet browsers. In the context of the experiment (during lockdown in June and July 2020), videoconference software were extensively used, and running audio and video streams requires many CPU cycles and memory usage. Therefore, participants might be biased as they used these programs in addition to performing their class workloads, in comparison to the general public who may use communication software sparsely and with limited multitasking. These software are known to have a high energy impact on energy consumption [Zhang et al.(2009)]. Therefore, we recommend better optimization for communication software, and switching to less machine-intensive encoding/decoding algorithms to alleviate the load and energy consumption on mobile and computing devices.

Finally, as the dependencies of mobile devices towards cloud computing is rising, the energy consumption of either is therefore close and inseparable. Most mobile devices use cloud-based services to perform their intended features: a smart speaker depends on cloud services to provide accurate and relevant answers to users, or a smart fridge might use cloud services to provide an up-to-date shopping list of missing items, and similar arguments for other devices such as smart TVs or connected security cameras. We recommend outlining this dependency in green feedback tools, in order to show the impact of the entire chain of software and hardware when users interact with their devices.

5.2 Perception of green feedback

Our first observation of the reaction of users to our green feedback tool is that it helps in raising awareness of the actual power consumption of devices and software. Participants either exaggerated or minimized the energy costs, and nearly half of them did not understand what the metrics mean. Our recommendation follows what we observed in

the first part of the survey, that metaphors, such as a traffic-light system, might be more relevant to users.

In addition, participants thought that power metrics were most useful to understand the impact of energy. Without more in-depth investigation with participants, it is not straightforward to conclude that power metrics are easier to understand, and contradicts with a study from 2009 on the usage of energy meters in homes [Kelsey and Gonzalez(2009)], that indicated users are more interested in the price metric rather than the power one. However, as our participants were computer savvy (computer science and engineering students) and young, the power metric might actually be more relevant for them, as this metric is common for overclocking, gaming or other hardware-related studies and activities. In contrast, a small minority of participants, 6.25%, placed the CO2 emissions metric as the most helpful one. We argue that CO2 emissions are seen as an abstract measure, and it is hard for users to grasp how many grams of CO2 emissions is dangerous for global warming. Instead, we recommend abstracting much of these metrics, and use green-related or sustainability metrics. For instance, the number of trees saved, or the estimated degrees lowered or raised by the energy consumption or savings, might have a better effect on users.

Participants were also more impacted by the visual chart than the metrics themselves. The charts in our visual feedback tool is plotted in real time and its Y axis scales to show the maximum plot value, and therefore might had an impact of the users' perception of power consumption. Charts provide a short-term historical view, and most importantly, a comparative indication of an increase or decrease of power consumption. We argue it is important to keep such visual charts as it provides an immediate feedback of whether the actions of users have a positive or negative impact on energy consumption.

Our most interesting finding is that a majority of participants did not change their software usage behavior. And from the few who stated a change, a majority of changes happened when doing less important tasks, and only a minority changed behavior when a spike of energy consumption was seen in the feedback tool. This might be explained by the given workload, and the identity of participants: university students working on a project. We plan on conducting a long-term experiment with diverse profiles, in order to confirm our initial findings.

Participants also miss the tools and the knowledge of what to do to change their behavior, even if they want to. In particular, participants know two dimensions of actions:

- **Temporal:** by reducing the usage time of software, which was the most declared action. However, this answer should be put in perspective of the majority of users who did not change their behavior, or state that the change was not for important tasks. Concretely, we argue that users will rarely reduce the usage of programs or devices, unless they're using them for time-passing entertainment.
- **Tooling:** by switching to a different equivalent program. The main challenge in switching software, is the availability of quality and functionally equivalent ones. Users need to know whether switching to a different program will lead to energy reductions. We argue that energy-related

software labeling is crucial, and that such labeling needs to be provided when installing software, and on every run (for example, in the form of a sign on software icons).

However, when we compared the actual power readings of the participants who stated they changed their behavior, we observe little or no variation in the energy consumption. In Figures 2 and 3, we outline the power consumption of some participants who stated they changed their behavior when they saw a spike in power consumption, or when doing less important tasks, respectively. The figures show the power consumption data and a cubic spline interpolation (with a smoothing parameter of 0.5). When applying a simple linear model over these numbers, we obtain the intercept and slope numbers seen in Table 9.

Participant	Intercept	Slope
Every time I saw a spike in energy consumption in the visual feedback window		
A	9.96	-0.001
B	6.58	0.0003
C	13.44	-0.00009
D	11.22	-0.0022
Only when I was doing less important tasks		
E	15.23	-0.0029
F	27.12	-0.0041
G	21.92	-0.0046

TABLE 9: Intercept and slope for the power consumption of participants in group A2

Overall, the slope in the linear model, although mostly negative (meaning a reduction of power consumption), is negligible. The only exception was with participant C who had a small reduction after a spike in the middle of the experiment. After checking the program logs, we find that when the spike happened, power consumption rose for a few seconds to 21 then 29 watts, far beyond the average TDP of the CPU model (an Intel i7-6500U). During this event, the participant was using Google Chrome and working on the workload in the online latex editor, Overleaf. As soon as the spike happened, we observe a gradual drop in consumption over the next five seconds, to around 7 watts. This coincides with a switch in software, where the participant opened our monitoring tool for a second, then opened Windows File Explorer, and kept using it for three minutes before switching back to Google Chrome.

Other participants did not apply behavioral changes, or did little changes in software usage with no lasting impact on power consumption. For example, participant A did adopt monitored behavioral changes in software usage (*i.e.*, did not change software, switch tabs, or change the active window). The biggest energy spike (between seconds 1500 and 1700) is associated with using Google Chrome (Overleaf and Google searches), which might be a more intensive work session. Participant B had one major spike around second 500 which is a direct result of accessing a specific web page on the course topic that contained text, images, a video and ads. The spike spanned for around 30 seconds, which might be due to the video auto-play. We did not observe any change in behavior specific to our feedback tool, as the user switched to Overleaf website (as expected from the course workflow) after the end of the power spike.

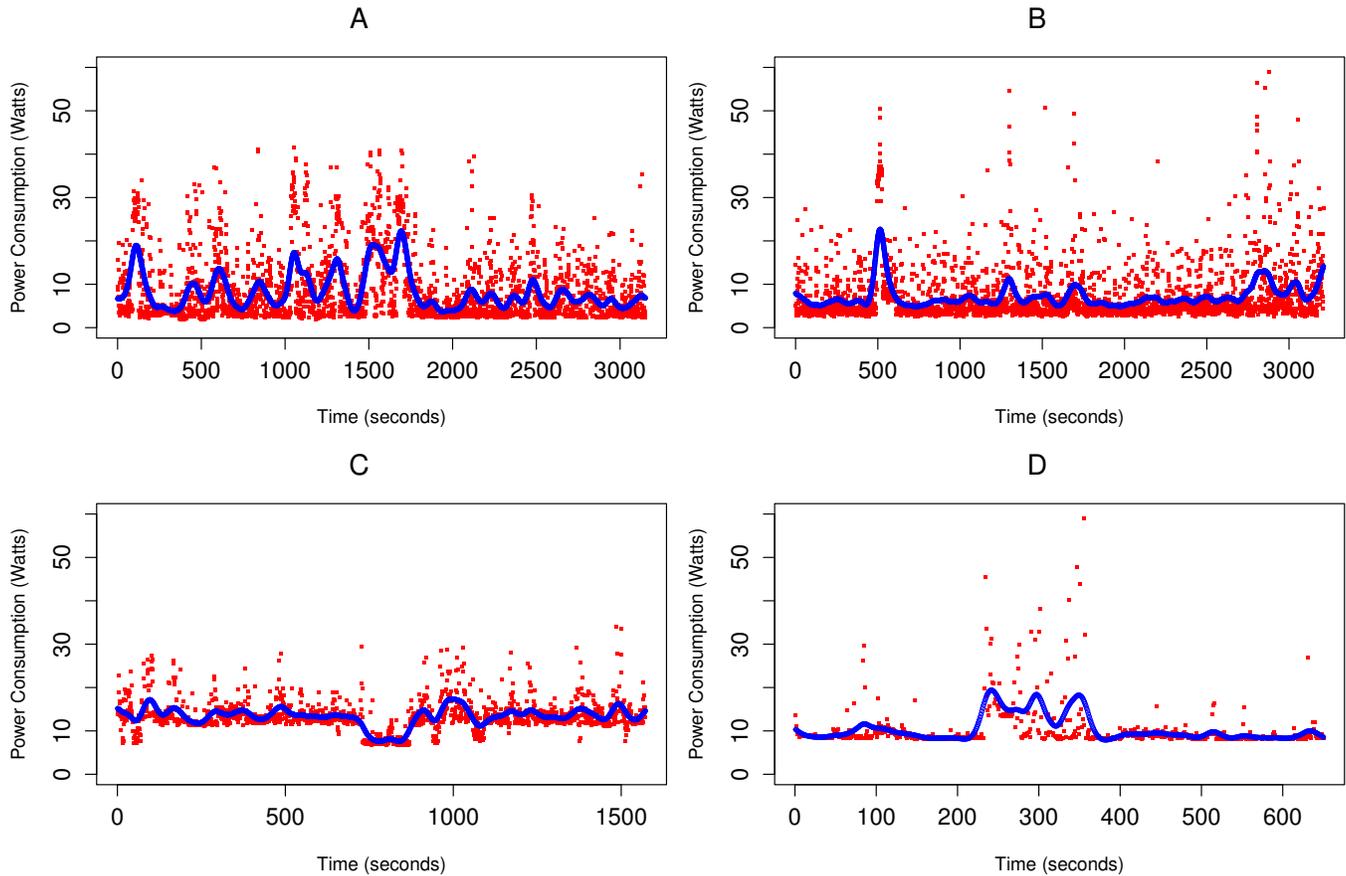


Fig. 2: Power consumption and cubic spline interpolation (red line) of participants in group A2 who stated a software change every time they saw a spike in energy consumption in the visual feedback window.

Participant D power spike is explained by opening a PDF thesis online for around 10 seconds, then participant closed the tab which resulted in a minor reduction of energy, and finally switched back to Windows Explorer.

In contrast, users who stated to change behavior when doing less important tasks, did not translate into power reductions. For instance, participant 3 has a consistent usage of software and that translates with consistent power consumption with few spikes. Participant F had a more intensive and diverse usage of his computer (including for checking email and WhatsApp), but we did not observe any behavioral change when not working on the course assignment. Some of the power spikes actually amounts to personal usage of the laptop, with spikes when using WhatsApp desktop application. Finally, participant G also has a consistent power consumption which is due to a continuous intensive usage of software for the assignment and for personal usage (desktop email client, WhatsApp on Google Chrome).

From these findings, we can conclude that participants, even when trying to change their behavior, fail to reduce their energy consumption. This might be due to little knowledge of what to do (which 58.33% of participants who wanted to change behavior didn't know how to do so). Or due to the actions applied, such as changing software or reducing usage time, didn't have the expected effect by users. Even users who changed behavior, did not correctly

or sufficiently apply the changes they state they did. For instance, participant D stated a change in software every time the tool showed a spike. However we did not observe any reasonable switch in software in the log data beyond the normal changes in the workload.

We argue that green feedback needs to be accompanied by runtime propositions of change, along with a sustainability guide. For instance, an up-to-date sustainability guide or documentation might propose a list of equivalent software for the most important tasks on a device (*e.g.*, internet browser, word processing, popular applications such as weather or music), along with an energy-related label or classification. In addition, the green feedback tool needs to propose, on the fly, recommendations on what do to reduce energy usage. These recommendations would be based on the user historical usage, the energy label of software, and the current energy consumption and/or price and CO2 emissions. Recommendations would be clear and optional, and if the user accepts the proposition, the system must apply a smooth, low-energy, and low-latency transition of software without data loss.

6 LESSONS LEARNED

From the field study and the survey, and our analysis, we draw the following lessons:

- **Users underestimate the energy consumption of their devices and software.** As we observed in our survey,

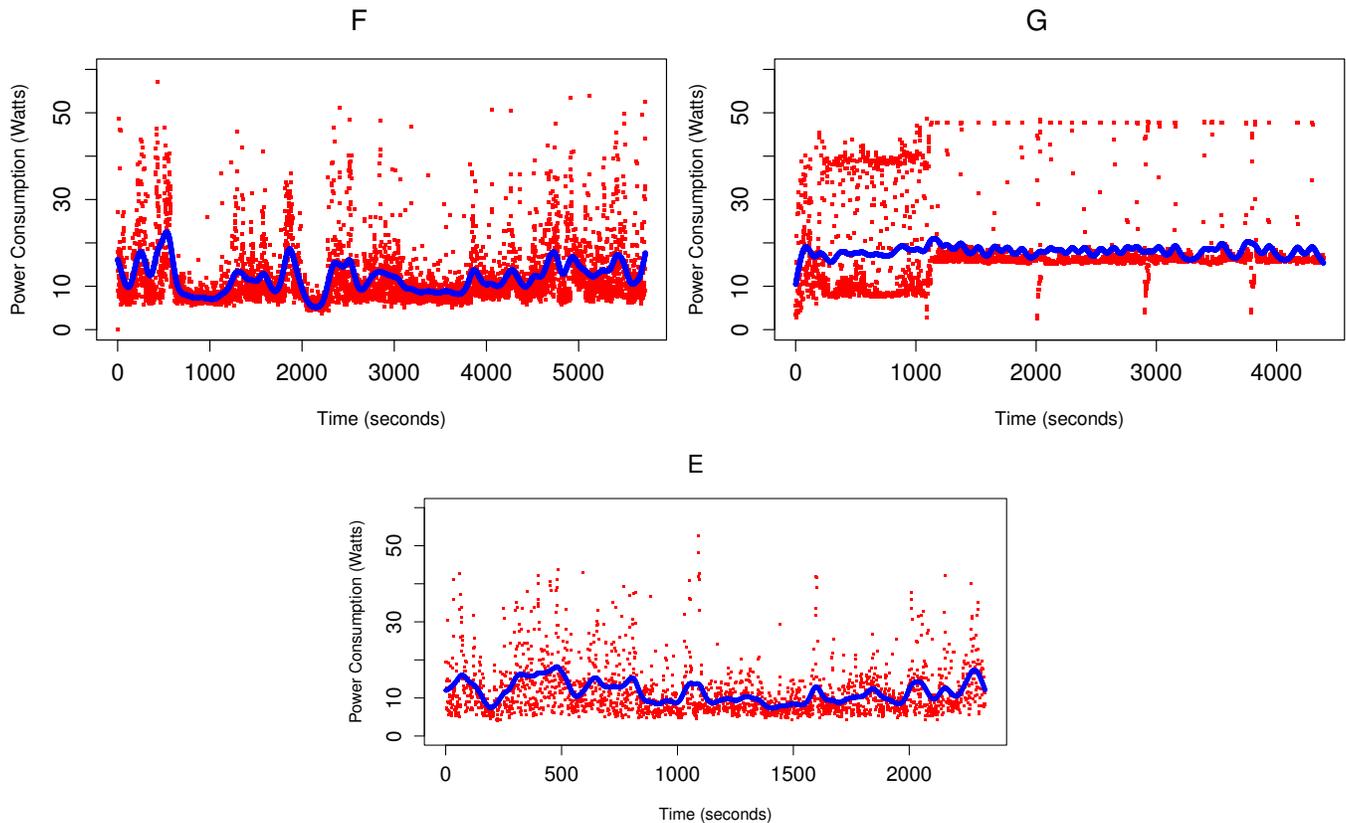


Fig. 3: Power consumption and cubic spline interpolation (red line) of participants in group A2 who stated a software change only when doing less important tasks.

users in the control group rated their energy session lower than users who saw their actual power consumption through our green feedback tool.

- **Users perceive the operating system as the most consuming software.** The OS is seen as the most power-consuming software in our experiments, for both the control and experiment groups. High or constant idle power consumption might explain this perception. But also how slow or sluggish systems can be, even though it might be because of other software, most notably background ones.
- **Programming and communication programs are perceived as high-consuming software.** For computing science and engineering students, sometimes using older devices, heavy-duty IDEs can be seen as slow and consuming resources, including power. The same applies for communication software, in particular videoconference programs that might stress the CPU and the device's resources.
- **Technical-savvy users understand the energy weight of the cloud infrastructure.** Most participants in our field study, acknowledge the bigger impact of the cloud infrastructure (*e.g.*, data centers, network) compared to individual end-user devices. However user awareness of the chain of impacts of these devices and software services towards the cloud infrastructure is yet to be properly understood.
- **Live green feedback helps in raising awareness of energy consumption.** Although we did not observe notice-

able changes in the overall power consumption between the control group A1 and the experiment group A2, we observed tentative behavioral changes for users in group A2 (with our green feedback tool). Participants wanted to change their behavior but had a few obstacles to do so.

- **Green feedback tools must be minimal and seamlessly integrated to avoid being distracted.** The majority of participants saw our green feedback window as distracting due to its nature (always on top of other windows, and could not be minimized or resized, even though it was small and only covered a limited space in the lower right-end of the screen). The distraction nature of the feedback window could not be scientifically assessed until after we conducted the experiments and collected participants' answers. We argue that green feedback needs to be transparently integrated into the operating system or software systems of end-user devices.
- **Metaphors and evolving charts are more useful than green metrics.** Specific green metrics, such as power consumption, electricity price, or CO₂ emissions, do not seem to have an effect on end users. However, charts indicating the evolution of power consumption had a better impact (participants changed behavior following power spikes in the chart). We also argue that metaphors, such as a traffic-light system or the number of trees saved, might lead to better awareness and behavioral change.
- **Users resist behavioral change when the workload is important, or the energy consumption is perceived as**

low. Many participants who tried to change their energy behavior, did so when doing less important tasks, or only when an important spike in energy occurred. This indicates that, depending on the workload, users will prioritize the tasks they're doing, over non-functional energy requirements. This might be due to the lack of knowledge of the cost of changing software or reducing usage, or other energy-efficiency actions.

- **Users lack the knowledge and tools to apply software behavioral changes.** A majority of participants who wanted to change their behavior failed to do so because they didn't know what to do. And the remainder who reduced software usage or switched software, did not apply those actions correctly or long enough for any noticeable energy savings. We argue that users need guidance and recommendations in order to know what actions are effective and simple to implement.

7 LIMITATIONS AND THREATS TO VALIDITY

We acknowledge there are several limitations and threats to validity to our field study. First, our sample included more educated (university students) and more technology savvy (computer science and engineering) participants. This limits the validity of our analysis and results as they only apply on this specific demographics (*i.e.*, computer professionals). However, as a first of its kind large field study, we argue the results show promising trends in end users, and can be the basis for wider scale studies on more demographic groups.

Second, our sample is mostly composed of males (79%), but the proportions are balanced between the control group A1 and the experiment group A2. Third, our workload, although representative of typical cloud-based usage of most software today, was executed in a classroom scenario. Participants (who agreed to participate to the experiment as it was not mandatory) had to work on a document that was later marked but the mark did not count on their course final grades. We did not apply random assignment or pre-test to construct our control and treatment groups (we assigned whole classes to either group). However, there are no major differences between our groups (A1 and A2), with a similar number of participants, average age, gender distribution and job profiles.

Then, as stated in the previous sections, we observe the need for more in-depth field studies, with additional software data to collect, a more comprehensive survey, and the need for one-to-one interviews with participants. In particular, we would like to conduct additional field studies to understand why participants who wanted, or stated, a behavioral change, failed to do so correctly or efficiently. In addition, the implementation of the architecture is limited to measuring CPU energy consumption (*i.e.*, no other hardware components were measured), and we used the standard electricity price of the public company of the users' geographic location (*i.e.*, electricity price was only tailored for the user's region and main utility provider, and not for the user specific contract or a different competing utility provider). As for future experiments, our tool can be extended to monitor the power consumption of additional hardware components, such as the RAM memory, the graphic card GPU, or the storage unit (hard disk, SSD).

The tool can also monitor the power consumption per application and monitor background processes and operating system services. Although we are aiming for monitoring additional components and software in future work, it is important to stress that end users might not find technically detailed energy data relevant or useful for behavioral changes. Users interact with active windows of software and might have difficulties associating energy consumption of an OS routine to their actions in using software. For our study in this paper, we limited data collection to a few components (the CPU) and software (applications running and active window), in order to focus on user behavior. However, additional field experiments and studies are necessary to fully analyze user understanding of software and energy consumption, including for other hardware components or background software services.

Furthermore, our study is limited to a specific workload environment (a class environment with a given assignment). We did not cover other software usages, such as entertainment usages (*e.g.*, watching a movie, or playing a video game), nor a work environment. Finally, our study focuses on individual energy consumption and behaviors, rather than a group of users, or a group of devices. For instance, software energy consumption is also a concern for industrial setups or factories. Although we think important energy saving could be achieved in these environments, our study do not cover them. Our study shows an interesting potential in raising awareness of the cost of software energy in users' devices, and a sizable number of participants were willing or tried to apply actions to mitigate their energy and ecological impact.

8 CONCLUSION

In this paper, we presented the results of a field study and survey we conducted in order to understand what users know about energy in devices and software, and how can green feedback raise their awareness and push for behavioral changes. We also outlined our recommendations and lessons learned. We studied the impact of awareness through live visual feedback using green metrics (power consumption, electricity price and CO2 emissions), and real-time evolving charts. Our main conclusion is that green feedback helps in raising awareness, but users lack the knowledge and tools to apply software behavioral changes. Therefore, short-term behavioral changes require previous awareness and guidance so users can change behavior seamlessly. In addition, we observe the importance of the workload, and the tasks when using connected devices and software, as users resist software behavioral change unless the task was less important or its perceived energy is low.

For future work, we intend to further investigate the impact of green feedback through additional forms and indicators, in particular using metaphors. We also aim to produce guides, listing of alternative software, and energy-related classifications and labels, and study their impact, along with green feedback, on software behavioral change. Finally, we intend to enrich our architecture with a machine-learning recommendation-based green feedback tool, to help users choose the best energy-efficient actions and drive behavioral change in their software usage. For instance, the tool would

automatically provide guidance and recommendations on alternative software to use for a specific task (e.g., switch to a different word processor, among the installed ones), or tweak the options of the same software, or other learned recommendations.

REFERENCES

- [Anderson and Lee(2016)] Kyle Anderson and SangHyun Lee. 2016. An empirically grounded model for simulating normative energy use feedback interventions. *Applied Energy* 173 (jul 2016), 272–282. <https://doi.org/10.1016/J.APENERGY.2016.04.063>
- [Ardito et al.(2015)] Luca Ardito, Giuseppe Procaccianti, Marco Torchiano, and Antonio Vetrò. 2015. Understanding green software development: A conceptual framework. *IT Professional* 17, 1 (2015), 44–50. <https://doi.org/10.1109/MITP.2015.16>
- [Bashroush et al.(2016)] Rabih Bashroush, Eoin Woods, and Adel Noureddine. 2016. Data Center Energy Demand: What Got Us Here Won't Get Us There. *{IEEE} Software* 33, 2 (2016), 18–21. <https://doi.org/10.1109/MS.2016.53>
- [Belkhir and Elmeligi(2018)] Lotfi Belkhir and Ahmed Elmeligi. 2018. Assessing ICT global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production* 177 (mar 2018), 448–463. <https://doi.org/10.1016/J.JCLEPRO.2017.12.239>
- [Buchanan et al.(2015)] Kathryn Buchanan, Riccardo Russo, and Ben Anderson. 2015. The question of energy reduction: The problem(s) with feedback. *Energy Policy* 77 (2015), 89–96. <https://doi.org/10.1016/j.enpol.2014.12.008>
- [Colmant et al.(2018)] Maxime Colmant, Romain Rouvoy, Mascha Kurpicz, Anita Sobe, Pascal Felber, and Lionel Seinturier. 2018. The next 700 CPU power models. *Journal of Systems and Software* 144 (oct 2018), 382–396. <https://doi.org/10.1016/j.jss.2018.07.001>
- [Darby(2001)] Sarah Darby. 2001. Making it Obvious: Designing Feedback into Energy Consumption. In *Energy Efficiency in Household Appliances and Lighting*. Springer Berlin Heidelberg, Berlin, Heidelberg, 685–696. https://doi.org/10.1007/978-3-642-56531-1_73
- [Evans(2011)] Dave Evans. 2011. *How the Next Evolution of the Internet Is Changing Everything*. White Paper. Cisco.
- [Fischer(2008)] Corinna Fischer. 2008. Feedback on household electricity consumption: A tool for saving energy? *Energy Efficiency* 1, 1 (2008), 79–104. <https://doi.org/10.1007/s12053-008-9009-7>
- [Francisco and Taylor(2019)] Abigail Francisco and John E. Taylor. 2019. Understanding citizen perspectives on open urban energy data through the development and testing of a community energy feedback system. *Applied Energy* 256 (2019), 113804. <https://doi.org/10.1016/j.apenergy.2019.113804>
- [ICTFOOTPRINT.eu(2015)] ICTFOOTPRINT.eu. 2015. ICT Carbon footprint. <https://ictfootprint.eu/fr/about/ict-carbon-footprint/ict-carbon-footprint>
- [Jagroep(2017)] Erik Jagroep. 2017. *Green Software Products*. Ph.D. Dissertation. Utrecht University.
- [Jain et al.(2013)] Rishree K. Jain, Rimas Gulbinas, John E. Taylor, and Patricia J. Culligan. 2013. Can social influence drive energy savings? Detecting the impact of social influence on the energy consumption behavior of networked users exposed to normative eco-feedback. *Energy and Buildings* 66 (nov 2013), 119–127. <https://doi.org/10.1016/J.ENBUILD.2013.06.029>
- [Kelsey and Gonzalez(2009)] C Kelsey and V Gonzalez. 2009. Understanding the use and adoption of home energy meters. In *Proceedings of El Congreso Latinoamericano de la Interaccion Humano-Computadora*. 64–71.
- [Konstantakopoulos et al.(2019)] Ioannis C. Konstantakopoulos, Andrew R. Barkan, Shiyang He, Tanya Veeravalli, Huihan Liu, and Costas Spanos. 2019. A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure. *Applied Energy* 237 (2019), 810 – 821. <https://doi.org/10.1016/j.apenergy.2018.12.065>
- [Level(2015)] EURECA project. 2015. *Report on impact analysis of green data centre procurement choices*. Technical Report. H2020 Data centre EURECA Project. 1–52 pages.
- [Lu(2018)] C. Lu. 2018. IoT-Enabled Adaptive Context-Aware and Playful Cyber-Physical System for Everyday Energy Savings. *IEEE Transactions on Human-Machine Systems* 48, 4 (2018), 380–391.
- [Manotas et al.(2016)] Irene Manotas, Christian Bird, Rui Zhang, David Shepherd, Ciera Jaspan, Caitlin Sadowski, Lori Pollock, and James Clause. 2016. An empirical study of practitioners' perspectives on green software engineering. *Proceedings - International Conference on Software Engineering* 14-22-May- (2016), 237–248. <https://doi.org/10.1145/2884781.2884810>
- [Noureddine et al.(2013a)] Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013a. A review of energy measurement approaches. *Operating Systems Review* 47, 3 (2013), 42–49. <https://doi.org/10.1145/2553070.2553077>
- [Noureddine et al.(2013b)] Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013b. A review of middleware approaches for energy management in distributed environments. *Software: Practice and Experience journal* 43, 9 (2013), 1071–1100. <https://doi.org/10.1002/spe.2139>
- [Noureddine et al.(2015)] Adel Noureddine, Romain Rouvoy, Lionel Seinturier, Adel Noureddine, Romain Rouvoy, Lionel Seinturier, and Monitoring Energy. 2015. Monitoring energy hotspots in software. *Automated Software Engineering* 22, 3 (oct 2015), 291–332. <https://doi.org/10.1007/s10515-014-0171-1>
- [Orgerie et al.(2014)] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. 2014. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *Comput. Surveys* 46, 4 (mar 2014), 1–31. <https://doi.org/10.1145/2532637>
- [Orland et al.(2014)] Brian Orland, Nilam Ram, Dean Lang, Kevin Houser, Nate Kling, and Michael Coccia. 2014. Saving energy in an office environment: A serious game intervention. *Energy and Buildings* 74 (2014), 43 – 52. <https://doi.org/10.1016/j.enbuild.2014.01.036>
- [Paone and Bacher(2018)] Antonio Paone and Jean Philippe Bacher. 2018. The impact of building occupant behavior on energy efficiency and methods to influence it: A review of the state of the art. *Energies* 11, 4 (2018). <https://doi.org/10.3390/en11040953>
- [Pinto and Castor(2017)] Gustavo Pinto and Fernando Castor. 2017. Energy Efficiency: A New Concern For Application Software Developers. *Commun. ACM* 60, 12 (nov 2017), 68–75. <https://doi.org/10.1145/3154384>
- [Sánchez et al.(2014)] L. Sánchez, I. EliceGUI, J. Cuesta, and L. Muñoz. 2014. On the energy savings achieved through an internet of things enabled smart city trial. In *2014 IEEE International Conference on Communications (ICC)*. 3836–3841.
- [Vassileva et al.(2012)] Iana Vassileva, Monica Odlare, Fredrik Wallin, and Erik Dahlquist. 2012. The impact of consumers' feedback preferences on domestic electricity consumption. *Applied Energy* 93 (may 2012), 575–582. <https://doi.org/10.1016/J.APENERGY.2011.12.067>
- [Wood and Newborough(2007)] G. Wood and M. Newborough. 2007. Energy-use information transfer for intelligent homes: Enabling energy conservation with central and local displays. *Energy and Buildings* 39, 4 (apr 2007), 495–503. <https://doi.org/10.1016/J.ENBUILD.2006.06.009>
- [Reeves et al.(2015)] Byron Reeves, James J. Cummings, James K. Scarborough, and Leo Yeykelis. 2015. Increasing Energy Efficiency With Entertainment Media: An Experimental and Field Test of the Influence of a Social Game on Performance of Energy Behaviors. *Environment and Behavior* 47, 1 (2015), 102–115. <https://doi.org/10.1177/0013916513506442>
- [Karlin et al.(2014)] Beth Karlin, Rebecca Ford, and Cassandra Squiers. Energy feedback technology: A review and taxonomy of products and platforms. *Energy Efficiency*, 7(3):377–399, 2014.
- [Zhang et al.(2009)] Jiucai Zhang, Dalei Wu, Song Ci, Haohong Wang, and Aggelos K Katsagelos. 2009. Power-aware mobile multimedia: a survey. *J. Commun.* 4, 9 (2009), 600–613.
- [Zhang et al.(2022)] Feng Zhang, Jidong Zhai, Xipeng Shen, Onur Mutlu, and Xiaoyong Du. 2009. POCLib: A High-Performance Framework for Enabling Near Orthogonal Processing on Compression. *IEEE Transactions on Parallel and Distributed Systems*. 33, 2 (2022), 459–475.
- [Noureddine et al.(2012)] Adel Noureddine, Aurelien Bourdon, Romain Rouvoy, and Lionel Seinturier. 2012. A preliminary study of the impact of software engineering on GreenIT. *Proceedings - First International Workshop on Green and Sustainable Software* 3-june- (2012), 21–27. 10.1109/GREENS.2012.6224251
- [Fieni et al.(2020)] Guillaume Fieni, Romain Rouvoy, and Lionel Seinturier. 2020. SmartWatts: Self-Calibrating Software-Defined Power Meter for Containers. *Proceedings - 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing* 11-14-may- (2020), 479–488. 10.1109/CCGrid49817.2020.00-45