2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)

GreenHub Farmer: Real-world data for Android Energy Mining

Hugo Matalonga[†], Bruno Cabral[*¶], Fernando Castor[§], Marco Couto[†‖]
Rui Pereira[†‖‡] Simão Melo de Sousa[‡**] and João Paulo Fernandes[*¶]

[*] *Universidade de Coimbra, Portugal* [†] *Universidade do Minho, Portugal*
[‡] *Universidade da Beira Interior, Portugal* [§] *Universidade Federal de Pernambuco, Brasil*
[¶] *CISUC, Portugal* [‖] *HASLab/INESC TEC, Portugal* [**] *LISP, Portugal*

hugo@hmatalonga.com, bcabral@dei.uc.pt, castor@cin.ufpe.br, marco.l.couto@inesctec.pt,
ruipereira@di.uminho.pt, desousa@di.ubi.pt, jpf@dei.uc.pt

***Abstract*—As mobile devices are supporting more and more of our daily activities, it is vital to widen their battery up-time as much as possible. In fact, according to the Wall Street Journal, 9/10 users suffer from low battery anxiety. The goal of our work is to understand how Android usage, apps, operating systems, hardware and user habits influence battery lifespan. Our strategy is to collect anonymous raw data from devices all over the world, through a mobile app, build and analyze a large-scale dataset containing real-world, day-to-day data, representative of user practices. So far, the dataset we collected includes 12 million+ (anonymous) data samples, across 900+ device brands and 5.000+ models. And, it keeps growing. The data we collect, which is publicly available and by different channels, is sufficiently heterogeneous for supporting studies with a wide range of focuses and research goals, thus opening the opportunity to inform and reshape user habits, and even influence the development of both hardware and software for mobile devices.**

I. Introduction

MOBILE devices have become one of our most used gadgets, with their battery life becoming of a high concern for users. In fact, battery life is traditionally known to be one of the major factors influencing consumer satisfaction [1], and was just recently pointed out, on top of usability, storage and durability, as the most important feature for smartphone owners [2]. Battery life is such a growing concern that it has been shown that 9 of 10 users suffer from low battery anxiety [3], and is under discussion as a potential clinical condition: *nomophobia*, the fear of being without your mobile phone, in the Diagnostic and Statistical Manual of Mental Disorders [4].

On the other end, developers are also very concerned with their application's battery life, as excessive battery consumption is one of the most common causes for bad app reviews in app stores [5], [6]. In fact, developers are aware of the battery consumption problem, and many times seek help in solving this, even if they rarely receive adequate advice [7]–[9]. Mobile brands have actually caught sight of this issue and have tried to offer help by publishing developer guides aimed at extending battery life[1][2][3].

Reducing the energy that is consumed by mobile devices is also an important problem from a sustainability point of view. Indeed, the billions of phones that are in use these days have a global massive environmental footprint, and our digital consumption (which includes but is not limited to mobile device usage) is bound to have a greater impact on global warming than the aviation industry [10].

Despite its importance, optimizing, or even analyzing energy consumption for mobile devices is a difficult and labor-intensive task for both users and/or developers.

For once, developers are using different monitoring tools [11]–[13] according to specific needs which often results in a non systematized procedure and context specific findings [13]–[15]. Monitoring the energy consumed by an application often results in extensive tests under several different scenarios and devices [16]–[18], both very time consuming and potentially requiring large initial investments. Indeed, even considering Android alone, this is already a heavily heterogeneous environment, as there exists thousands of potential combinations among manufacturers, devices, operating systems, features, hardware components and apps, for example.

For users, understanding the energy consumption of their devices is an even harder exercise. For once, their knowledge regarding the hardware behavior is limited to their own devices, and without the proper tools and skills they cannot compare the energy behavior of their apps with others. Moreover,

We thank Ruben Alves and Pedro Carvalho for their contributions to GreenHub. This research was partially funded by: the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-016718; Operação Centro-01-0145-FEDER-000019 C4 Centro de Competências em Cloud Computing, co-financed by the Programa Operacional Regional do Centro (CENTRO 2020), through the Sistema de Apoio à Investigação Científica e Tecnológica Programas Integrados de IC&DT; INES 2.0, (FACEPE PRONEX APQ 0388-1.03/14 and APQ-0399-1.03/17, and CNPq 465614/2014-0), FACEPE APQ-0839-1.03/14, and CNPq 304220/ 2017-5.

[1] https://developer.android.com/topic/performance/power/

[2] https://developer.android.com/guide/topics/location/battery

[3] https://developer.android.com/docs/quality-guidelines/building-for-billions-battery-consumption

2574-3864/19/$31.00 ©2019 IEEE
DOI 10.1109/MSR.2019.00034

different usage contexts of the same app (e.g., within different OS versions and with different hardware components switched on) results in different energy behaviors, and this has to be taken into account when performing any comparison.

In this paper, we present a large dataset which is representative of real-world day-to-day usage of Android devices. Our dataset entries include information such as active sensors, memory usage, battery voltage and temperature, running applications, model and manufacturer, network details, etc,. This raw data was obtained by continuous crowd-sourcing through a mobile application. It is worth noting that all our data is publicly available, while maintaining the anonymity and privacy of all its users. Indeed, it is impossible to associate any data with the user who originated it. Thus far, our dataset includes unique 12 million+ samples, from 900+ different brands and 5,000+ models, across 160 countries.

This dataset was gathered within the GreenHub initiative[4], a collaborative approach to energy consumption analysis within Android. Our vision is to use the gathered data on the usage of mobile devices and application execution to help analyze and identify opportunities to optimize energy consumption in Android devices, both for developers and users. Indeed, we expect that useful information can be mined from the dataset as to help influence users in adopting more energy efficient behaviors and to provide developers with indications of how efficient their application is and how it compares to others.

In the case of developers, this is expected to trigger further analyses which are beyond the dataset itself. These may explore the potential energy gains that have, e.g., been proposed in the context of location services [19], contrast [20], color scheme [20], [21], data structure [22]–[25], programming language [26]–[29], network usage [15], and API [17] usage.

The remainder of this paper will describe: the developed infrastructure (Section II) for the data collection, dataset, and a data-query prototyper; possible research directions for developers and users (Section III); and finally the conclusions of this paper (Section IV).

II. Collaboratively Collecting and Sharing Data

The GreenHub initiative is committed to provide the means to support a symbiotic relationship with the mobile community. In fact, the success of the initiative is dependent on its data, and to keep such data coming in, we plan to give back to the community in concrete and valuable ways.

The development was inspired by Carat [30], an Android app which collects data regarding apps running on a device, and uses it to provide battery-saving recommendations, such as when to close an app. Initially, we forked its open-source code to take advantage of the data collection and storage mechanisms. We also updated its data model to consider more details on modern devices (NFC, Flashlight, etc,.).

In the same spirit, not only is our tool able to be freely used by others, but it is also entirely open-source. In contrast, all our collected data is permanently and publicly available, to strongly encourage and help others in collaborating, inspecting and/or reusing any artifact that we have developed or collected.

The initiative relies on a multi-component technological platform. An overview of our platform's architecture is shown in Figure II.1. It is composed of three open-source modules[5], which will be described in detail in the following sub-sections.

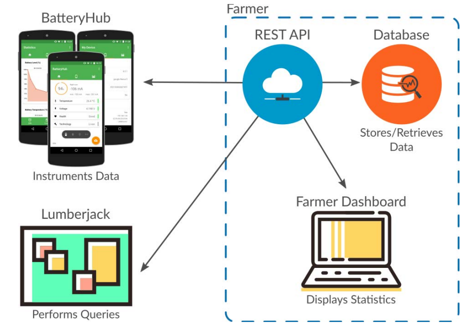

Fig. II.1. GreenHub platform architecture

A. BatteryHub - Data Collection

BatteryHub is a mobile application for Android which is available at Google's Play Store[6]. BatteryHub tracks system event broadcasts such as changes to the battery's state, and when such an event occurs, it obtains a sample of the device's current state. Each sample characterizes a wide range of aspects[7] that may affect battery usage, such as sensor usage, temperature or what applications are in use. BatteryHub either uses the official Android SDK or custom implementations for universal device compatibility support, and periodically communicates with the server application (over HTTP) to upload, and afterwards remove, the locally stored samples.

We would like to mention that the data collected from each user is made anonymous by construction. Indeed, it is associated with a random unique identifier and no personal information, such as phone number, location, or IMEI, is collected. This mean that it is (strictly) not possible to identify any user of BatteryHub, nor is it possible to associate any data with the user who originated it.

As to start giving back to our users as early as possible, BatteryHub already provides detailed information on their device. Currently, it indicates: i) the electric current level, temperature, voltage levels in a given period, and ii) model specifications, network information, memory usage, or storage details. Information in i) is re-actively updated when the battery's state changes, and in ii) when a system event occurs. A fully featured task manager is included, and the application also provides interactive charts throughout different time periods showing the changes on the different aspects of the battery.

The app allows for configurable alerts, e.g. when the battery reaches a certain temperature, and our plan is to use Battery-

[4] https://greenhubproject.org/

[5] GreenHub Code Repository: https://github.com/greenhub-project

[6] BatteryHub App: https://play.google.com/store/apps/details?id=com.hmatalonga.greenhub

[7] The concrete aspects will be described in detail in Section II-B and are publicly available in the application's terms and conditions and privacy policy.

Hub to give suggestions to users, based on their profiles, on how to reduce the energy consumption of their device.

Having deployed BatteryHub, our main challenge in constructing the dataset was the acquisition of a large user base. For this, we were helped by our institutions and their media outlets to bring attention to and attract the general audience. Our strategy achieved circa 50 dedicated publications from national and institutional venues, through news[8], magazines[9], newspapers[10], and radio shows[11][12], in Portugal and Brazil alone. This attracted an initial large group of users to the application, naturally propagating outside of the host countries.

B. Farmer - Dataset

As of December 2018, we have collected over 12.22 million data samples from 36,933 unique mobile devices, which represent 36 Android versions, 964 brands and 5665 models.

In order to store the collected data, we have implemented Farmer, a PHP server application, built on top of the Laravel framework[13]. Each sample is received as a JSON formatted string which is then processed in a queue job to be deconstructed and correctly mapped within a MariaDB relational database. Currently, the complete database is available as a MariaDB dump file[14], whose simplified data model can be seen in Figure II.2. We present a more detailed description (excluding primary and foreign keys) of our database tables, including information on the attribute's name, type, and an example of each in Table II-B.

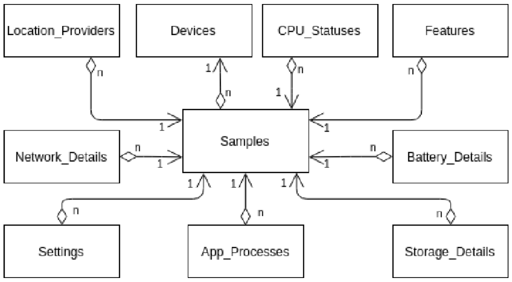

Fig. II.2. Entity Relationship diagram of the GreenHub dataset

By periodically executing a cron job, Farmer generates a dataset snapshot of the entire repository. As of December 2018, this resulted in a zip file[15] of 5.2GB, containing a CSV file for each dataset table.Farmer also includes a web dashboard interface[16] providing users access to up-to-date statistics, metrics, and charts of all the contained data.

TABLE I
DETAILS ON THE GREENHUB DATASET TABLES

Attribute Name	Type	Example
	App_Processes	
name	varchar	”com.facebook.katana”
application_label	varchar	”Facebook”
is_system_app	tinyInt	0
importance	varchar	”Service”
version_name	varchar	”8.2.0”
version_code	int	802000871
installation_package	varchar	com.android.vending
	Battery_Details	
charger	varchar	”unplugged”
health	varchar	”Good”
voltage	decimal	4.03
temperature	decimal	29.20
	Cpu_Statuses	
usage	decimal	0.03
up_time	bigInt	409480
sleep_time	bigInt	141369
	Devices	
model	varchar	”Nexus”
manufacturer	varchar	”LGE”
brand	varchar	”google”
product	varchar	”hammerhead”
os_version	varchar	”6.0.1”
kernel_version	varchar	”3.4.0-gcf10b7e”
is_root	tinyInt	0
	Network_Details	
network_type	varchar	”WIFI”
mobile_network_type	varchar	”lte”
mobile_data_status	varchar	”connected”
mobile_data_activity	varchar	”inout”
roaming_enabled	tinyInt	0
wifi_status	varchar	”enabled”
wifi_signal_strength	int	-71
wifi_link_speed	int	39
wifi_ap_status	varchar	”disabled”
network_operator	varchar	”verizon”
sim_operator	varchar	”unknown”
mcc	varchar	”311”
mnc	varchar	”480”
	Samples	
timestamp	timestamp	2017-10-08
app_version	int	11
database_version	int	3
battery_state	varchar	”Charging”
battery_level	decimal	0.90
memory_active	int	505296
memory_inactive	int	502392
memory_free	int	1442060
memory_user	int	60724
triggered_by	varchar	”android.intent.action.BATTERY_CHANGED”
network_status	varchar	”lte”
screen_brightness	int	-1
screen_on	tinyInt	1
timezone	varchar	”America/Chicago”
country_code	varchar	”us”
	Settings	
bluetooth_enabled	tinyInt	0
location_enabled	tinyInt	1
power_saver_enabled	tinyInt	0
flashlight_enabled	tinyInt	0
nfc_enabled	tinyInt	1
unknown_sources	tinyInt	0
developer_mode	tinyInt	0
	Storage_Details	
free	int	3922
total	int	9634
free_external	int	3922
total_external	int	9634
free_system	int	637
total_system	int	3390
free_secondary	int	0
total_secondary	int	0

Finally, an API within Farmer provides real-time selective access to the data repository: e.g., one may query all samples for a given brand or OS version. Since the API is designed according to the REST methodology, this allows us to incre-

[8] www2.cin.ufpe.br/site/lerNoticia.php?s=1&c=94&id=1697

[9] www.visao.sapo.pt/actualidade/sociedade/2017-10-11-Bateria-do-telemovel-invista-agora-para-poupar-depois

[10] https://www.publico.pt/2017/10/09/tecnologia/noticia/desenvolvida-aplicacao-para-poupar-bateria-de-dispositivos-moveis-1788153

[11] www.90segundosdeciencia.pt/episodes/ep-443-joao-paulo-fernandes/

[12] www.rtp.pt/play/p2063/e342304/ponto-de-partida

[13] https://laravel.com/

[14] GreenHub MariaDB database: https://farmer.greenhubproject.org/storage/greenhubfarmer_dump.sql

[15] GreenHub dataset ZIP: https://farmer.greenhubproject.org/storage/dec_2018/dataset_2018.zip

[16] Farmer Dashboard: https://greenhub.di.ubi.pt/

mentally add new data models to be reflected within the API itself as the data protocol evolves over time.

Farmer and BatteryHub make use of our API to communicate with each other, to send and store data, which also ensures an abstraction layer and added security to access the data repository. An example of a SQL query on our dataset to calculate the average temperature and voltage when the devices have Wifi enabled is shown in the following code snippet:

```
SELECT avg(temperature), avg(voltage)
FROM samples S, battery_details B, network_details N
WHERE S.id = B.sample_id and S.id = N.sample_id and
N.wifi_status = "enabled";
```

C. Lumberjack - Prototyping queries

Lumberjack is a command-line application interface allowing users to interact with the API. A user can perform flexible on-demand queries to the data repository, to support quick prototyping of data queries applying different filters and parameters. Furthermore, users can quickly fetch subsets of the data, without needing to download the entire dataset snapshot. An example of a Lumberjack query to obtain the list of Google brand devices can be seen in the following command:

```
$ greenhub lumberjack devices brand:google \
>   -o googleDevices.json
```

III. Research Directions

For developers, analyzing and identifying optimization opportunities is difficult, error-prone, and labor-intensive. Often times, they have to extensively test their products in countless scenarios and environments in order to understand how, and how much, energy is consumed. Also, such an analysis would require a large initial investment in devices and a large effort in developer time, and still, would not allow for an energy consumption comparison between similar apps. Furthermore, almost all mobile Operating System have severe battery consumption control policies, ”badly behaved” apps will get shutdown, almost instantly, when running in background.

Using the GreenHub dataset, one can try to answer app or system **developer** specific questions, such as:
1) How does my application impact battery consumption?
2) How has it evolved throughout different versions?
3) How does such impact vary considering different environments (operative system + installed/running applications)?
4) How does the consumption pattern of my application compare to other applications with similar goals?
5) Is the energy consumption of the application different for distinct users?
6) How are the OS policies for battery management influencing the up-time of my application?

For mobile users, it is arguably even harder to understand how energy is consumed on their devices, and in turn, how can it be optimized. While some users might have the common sense of deactivating sensors and applications when not in use, very few have the technical background or tools to perform a real analysis on their smartphone’s energy consumption, and even fewer will be able to depict the necessary optimization strategies, putting the data from such analysis to good use. Several Operating Systems have the necessary tools for the user to partially comprehend how the battery is drained and what apps influence it. But, it is impossible for each user individually to learn as much, and as quickly, as a collaborative approach, such as GreenHub, is able to provide for. Also, some questions would actually be impossible to answer without cross-analyzing data from multiple users and devices.

The GreenHub dataset helps to address **mobile user** specific research questions such as:
7) How does my battery consumption compares to the ones from other users with similar utilization profiles?
8) Considering other usage profiles, what can I reproduce to have a more energy efficient profile?
9) What is the impact on battery consumption of sensor usage?
10) Should I avoid running certain simultaneous applications?
11) Which of my applications are energy inefficient?
12) Could I replace some of the applications I use by more energy-efficient applications with the same functionalities?
13) What is the expected lifetime of my device’s battery considering my usage pattern?

To the best of our knowledge, Farmer has/is already supported/supporting the following studies:
• a characterization of the charging/discharging rates through different OS versions and manufacturer brands, which we ourselves are trying to accomplish;
• the quantification of the average impact of mobile games on smart phone battery consumption, and the identification of the most battery draining features of these games, conducted by Fernando Castor et al. at Federal University of Pernambuco;
• correlating the user-perceived and the actual battery-usage of mobile applications regarding their utilization profile, conducted by Ivan Machado et al. at Federal University of Bahia.

While these studies show some of the value of Farmer, we believe that there is significant room for further exploration. The large amount of data and features makes this dataset particularly interesting for usage with Machine Learning and AI techniques. Nevertheless, during future exploration of the dataset, we expect its limitations to become more evident.

IV. Conclusions

This paper presents the results of our GreenHub initiative to gather a large dataset called GreenHub Farmer. This dataset contains 12million+ samples of real-world usage crowd-sourced over 900+ different brands of Android devices across 160 countries. This data is both representative, and publicly available, allowing a collaborative approach to use our gathered mobile application and usage data to help analyze and identify opportunities to optimize energy consumption in Android devices, both for developers and users.

We aim to preserve a sustained movement towards extracting useful information from the collected data. For this, we invite other researchers and developers to both analyze and contribute to our dataset and the GreenHub initiative.

References

[1] A. Thorwart and D. O'Neill, "Camera and Battery Features Continue to Drive Consumer Satisfaction of Smartphones in US," https://www.prnewswire.com/news-releases/camera-and-battery-features-continue-to-drive-consumer-satisfaction-of-smartphones-in-us-300466220.html, 2017, last visit: 2019-02-06.

[2] "The most wanted smartphone features," https://www.statista.com/chart/5995/the-most-wanted-smartphone-features, accessed: 2018-01-24.

[3] T. Mickle, "Your phone is almost out of battery. remain calm. call a doctor." https://www.wsj.com/articles/your-phone-is-almost-out-of-battery-remain-calm-call-a-doctor-1525449283, 2018, last visit: 2019-02-05.

[4] N. L. Bragazzi and G. Del Puente, "A proposal for including nomophobia in the new dsm-v," *Psychology research and behavior management*, vol. 7, p. 155, 2014.

[5] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why people hate your app: Making sense of user feedback in a mobile app store," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1276–1284.

[6] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, "What do mobile app users complain about?" *IEEE Software*, vol. 32, no. 3, pp. 70–77, 2015.

[7] G. Pinto, F. Castor, and Y. D. Liu, "Mining questions about software energy consumption," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 22–31.

[8] I. Manotas, C. Bird, R. Zhang, D. Shepherd, C. Jaspan, C. Sadowski, L. Pollock, and J. Clause, "An empirical study of practitioners' perspectives on green software engineering," in *International Conference on Software Engineering (ICSE), 2016 IEEE/ACM 38th*. IEEE, 2016, pp. 237–248.

[9] C. Pang, A. Hindle, B. Adams, and A. E. Hassan, "What do programmers know about software energy consumption?" *IEEE Software*, vol. 33, no. 3, pp. 83–89, 2016.

[10] "Our phones and gadgets are now endangering the planet," https://www.theguardian.com/commentisfree/2018/jul/17/internet-climate-carbon-footprint-data-centres, accessed: 2018-01-24.

[11] D. D. Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, and A. D. Lucia, "Petra: A software-based tool for estimating the energy profile of android applications," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, May 2017, pp. 3–6.

[12] Y. Hu, J. Yan, D. Yan, Q. Lu, and J. Yan, "Lightweight energy consumption analysis and prediction for android applications," *Science of Computer Programming*, 2017.

[13] L. Cruz and R. Abreu, "Performance-based guidelines for energy efficient mobile applications," in *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*, ser. MOBILESoft '17. IEEE Press, 2017, pp. 46–57.

[14] D. Li and W. G. J. Halfond, "An investigation into energy-saving programming practices for android smartphone app development," in *Proceedings of the 3rd International Workshop on Green and Sustainable Software*, ser. GREENS 2014. New York, NY, USA: ACM, 2014, pp. 46–53. [Online]. Available: http://doi.acm.org/10.1145/2593743.2593750

[15] D. Li, Y. Lyu, J. Gui, and W. G. J. Halfond, "Automated energy optimization of http requests for mobile applications," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16. ACM, 2016, pp. 249–260.

[16] D. Li, S. Hao, W. G. Halfond, and R. Govindan, "Calculating source line level energy information for android applications," in *Proceedings of the 2013 International Symposium on Software Testing and Analysis*. ACM, 2013, pp. 78–89.

[17] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, "Mining energy-greedy api usage patterns in android apps: an empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 2–11.

[18] R. Jabbarvand, A. Sadeghi, J. Garcia, S. Malek, and P. Ammann, "Ecodroid: An approach for energy-based ranking of android apps," in *Proceedings of 4th International Workshop on Green and Sustainable Software*, ser. GREENS '15. IEEE Press, 2015, pp. 8–14.

[19] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 285–298.

[20] M. Linares-Vásquez, G. Bavota, C. E. B. Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, "Optimizing energy consumption of guis in android apps: a multi-objective approach," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 2015, pp. 143–154.

[21] M. Wan, Y. Jin, D. Li, J. Gui, S. Mahajan, and W. G. Halfond, "Detecting display energy hotspots in android apps," *Software Testing, Verification and Reliability*, vol. 27, no. 6, p. e1635, 2017.

[22] S. Hasan, Z. King, M. Hafiz, M. Sayagh, B. Adams, and A. Hindle, "Energy profiles of java collections classes," in *Proceedings of the 38th International Conference on Software Engineering*. ACM, 2016, pp. 225–236.

[23] R. Pereira, P. Simão, J. Cunha, and J. Saraiva, "jStanley: Placing a Green Thumb on Java Collections," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ser. ASE 2018. ACM, 2018, pp. 856–859.

[24] G. Pinto, K. Liu, F. Castor, and Y. D. Liu, "A comprehensive study on the energy efficiency of java's thread-safe collections," in *2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016, Raleigh, NC, USA, October 2-7, 2016*, 2016, pp. 20–31.

[25] R. Pereira, M. Couto, J. Saraiva, J. Cunha, and J. P. Fernandes, "The Influence of the Java Collection Framework on Overall Energy Consumption," in *Proceedings of the 5th International Workshop on Green and Sustainable Software*, ser. GREENS '16. ACM, 2016, pp. 15–21.

[26] W. Oliveira, R. Oliveira, and F. Castor, "A study on the energy consumption of android app development approaches," in *Proceedings of the 14th International Conference on Mining Software Repositories*. IEEE Press, 2017, pp. 42–52.

[27] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, and J. Saraiva, "Energy efficiency across programming languages: How do energy, time, and memory relate?" in *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, ser. SLE 2017. ACM, 2017, pp. 256–267.

[28] M. Couto, R. Pereira, F. Ribeiro, R. Rua, and J. Saraiva, "Towards a green ranking for programming languages," in *Proceedings of the 21st Brazilian Symposium on Programming Languages*, ser. SBLP 2017. ACM, 2017, pp. 7:1–7:8, best Paper.

[29] L. G. Lima, G. Melfe, F. Soares-Neto, P. Lieuthier, J. P. Fernandes, and F. Castor, "Haskell in Green Land: Analyzing the Energy Behavior of a Purely Functional Language," in *Proceedings of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER'2016)*. IEEE, 2016, pp. 517–528.

[30] A. J. Oliner, A. P. Iyer, I. Stoica, E. Lagerspetz, and S. Tarkoma, "Carat: Collaborative energy diagnosis for mobile devices," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys '13, Roma, Italy, November 11-15, 2013*. ACM, 2013, pp. 10:1–10:14.