Too Big to Mail: On the Way to Publish Large-scale Mobile Analytics Data

Ella Peltonen*, Eemil Lagerspetz*, Petteri Nurmi*†, Sasu Tarkoma*†
†Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki
*Department of Computer Science, University of Helsinki, PO Box 64, FI-00014, University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi

***Abstract*—The Carat project started in 2012 has collected over 1.5 TB of data from over 850,000 mobile users all over the world. The project uses Apache Thrift to transmit data, and Apache Spark to run data analysis tasks, and the gist of the Carat analysis method has been published. While the Carat application code is open source, the data is much harder to share because of its size and privacy concerns. This paper outlines the challenges in sharing such a large-scale dataset with detailed information about smart devices, applications, and their users, and presents some solutions to these challenges.**

***Index Terms*—Big Data, Mobile Analytics, Energy-awareness**

I. Introduction

Since its beginning in 2012, the Carat mobile application has collected data from 853,886 Android and iOS devices worldwide. The datasets include, for example, names and status of running and installed applications, current system settings, information of networking and system state, and so on. From a research point of view, the data has benefited analysis of energy-anomalous applications [8], system settings and subsystem variables [9], [11], malware applications [6], [13], and user behavior [2]. The Carat application goes further than simply collecting data. It also gives actionable energy-saving recommendations to the users. Indeed, further research could be conducted on the data to extend mobile user's experience of battery life, suggest better settings configurations, and discover malware in the wild. However, maintaining a large-scale system such as Carat places considerable strain on a research team.

In order to enable open access to the data, or a subset of it, for mobile application developers and other researchers, we face several challenges. How the data, which in its raw format will soon be over 2 TB, could physically made available? How to guarantee user privacy, so that the users of Carat are protected? How to guarantee that there are no legal or ethical issues in sharing the data, especially as it contains information of hundreds of thousands of applications and several hundred different device models?

The present paper discusses these challenges, not all of them yet fully solved, and discusses some solutions to consider in the near future when releasing larger portions of the Carat dataset. We also consider the development of a search-based data access API that the developers and the researcher community could take advantage of instead of a large raw dataset, and the merits and drawbacks of such an API.

If and when the data can be shared, mobile application developers could have a way to follow in real time their application's performance and energy efficiency. Manufacturers would be able to detect and mitigate broken hardware and software configurations. In terms of research, sharing the data would enable wider understanding of smartphone data issues in the wild across the many device models and usage patterns. Regular users of Carat would be able to gain a deeper understanding of the workings of their devices, and how they compare to others in the global market.

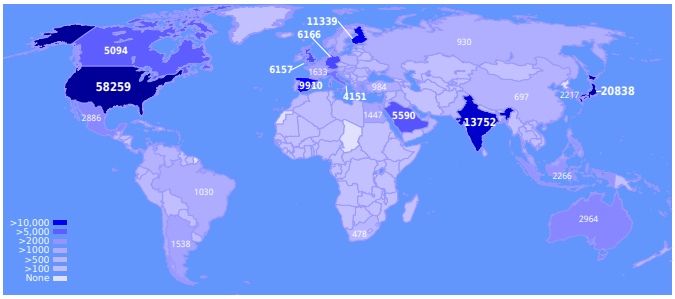

Fig. 1. Distribution of the Carat users worldwide (Android).

At the time of writing, a number of subsets of Carat data are provided by the NODES group at the University of Helsinki, and collaborating universities: the system settings and subsystem variables data has been published together with the related article [10], but without user identifiers or timestamps. The Carat website offers a set of statistically interesting visualizations based on the data for the distribution of device models, and top 200 most energy-intensive applications, and the Carat app itself gives a summary of basic statistics from the community. Carat data has not been shared in a large-scale way outside of the University of Helsinki and the University of California at Berkeley, where the project originally started. This paper discusses methods how to share an ever growing set of statistics and data from the Carat project.

II. Related Work

Surely some kind of mobile application data sets have been collected from the beginning of the mobile device era, but only limited number of these have ever published. Device Analyzer [14] has collected a large set of mobile device data, but the project published only samples with hashed application names, so it has only few or not at all use for developers. If accessed the data, it comes in large chunks impossible to

handle without dedicated software. We aim to make the data as easy to access as possible, with minimum risks for all the parties involved. Crowdsignals.io [1] is a new campaign to collect mobile data, but in its beginning and free for only crowdfunding participants.

However, few large-scale mobile usage datasets have been shared to date. This may be because of privacy concerns, since already a small number of fields can be used to glean information about the identity of the user [1]. Even the battery level can be used to uniquely identify website visitors [7] and the Wi-Fi signal strength to find out location [4]. Large file sizes are also an obstacle to sharing rich datasets including hundreds of thousands of devices.

Differential privacy [3] has been proposed as a solution for constructing APIs that allow access to large-scale datasets without exposing user identities. However, applying differential privacy to off-the-shelf algorithms can result in poor result quality [5]. However, differential privacy is attractive when we know the statistics that will be shared. In that case, implementation of differential privacy can be relatively lightweight [12].

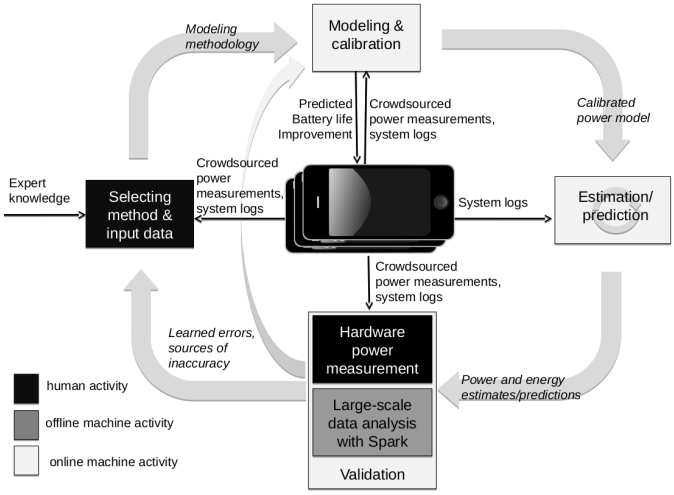

Fig. 2. The Carat feedback loop uses crowdsourced energy measurements and application activity logs, and its power profiles are validated with large-scale data analysis and hardware energy measurement.

III. Carat dataset

In the time of writing, the Carat application has been installed to 853,886 devices from 2012 until now. Fig. 1 shows the distribution of Carat's Android users in the world. Over 200 countries have Carat users, and the largest populations are in the US, Japan, India, and Finland. The application code for Android and iOS is available as open source on Github [2], and the application is available in Google Play and the App Store. The Carat analysis system collects the data sent by users and gives them energy saving recommendations. The analysis is run in Amazon EC2 with 10 memory-intensive virtual machines with 8 cores and 61 GB RAM each. The data is stored in Amazon S3 as Java/Scala objects within Spark RDD's that are readable in both Hadoop and Spark [15]. In binary format, the total data size of the data is around 1.5 TB, and much larger in text format. Monetary costs of running the analysis and storing the data to Amazon amount to some hundreds of dollars per month for the live system, not including Carat data research code also run in Amazon EC2.

Fig. 2 presents the current processing loop of the Carat system, including the data gathering from the devices, modeling and prediction phases, calibration and validation based on both data and laboratory measurements, and feedback loop back to the users and their devices. Open access to the parts of this process, could bring benefits for the stakeholders ranging from developers to researchers. Our analysis is implemented in Spark 2.0.1, and we also aim to publish parts of our Spark code base as open source.

The Carat application takes a sample every time the battery level changes by 1%, charging or discharging. This provides an energy-efficient method for data gathering, but in some cases, for example, when device lies in the power saving mode, we do not get samples. The Carat app records the following fields from the mobile devices, and does not collect any personally identifying information. Either do we not collect location or any location related-information, expect country and networking codes that reveal the origin of the data samples in country level. Note that some of the fields are platform-specific, for example, only available on Android devices:

a) Registration: sent when Carat is installed or the OS is updated. It includes a hash-based user identifier, timestamp, model code, OS version, distribution, and kernel version.

b) Sample: contains a hash-based user identifier, timestamp, current battery level in percent, battery state (charging/discharging), and a list of running applications.

c) Android only: includes screen brightness, CPU usage, uptime, sleeptime, screen on/off, developer mode, unknown application sources, and timezone. From certain devices we also get Bluetooth and location status. On iOS, we get whether power-saving mode is on or off.

d) Application: includes fields recorded for every application: process name, priority (background, foreground, etc), human-readable name, version code, version name, installation source, and application signatures.

e) Memory info: contains bytes of wired, active, inactive, free, and user memory. We also get the amount of free and used storage space, internal and SD card, from Android devices.

f) Battery info: contains information of charger type, battery health, voltage, temperature, technology, and capacity.

g) Network info: contains the network type, mobile data status and activity, roaming and Wi-Fi status, Wi-Fi signal strength, and Wi-Fi link speed. Fields available only on certain models include Wi-Fi AP status, sim operator, network operator, mobile country code (MCC), and mobile network code (MNC). From iOS devices we also get the total amount of Wi-Fi and mobile data sent and received.

[1] http://crowdsignals.io/

[2] https://github.com/carat-project/carat/

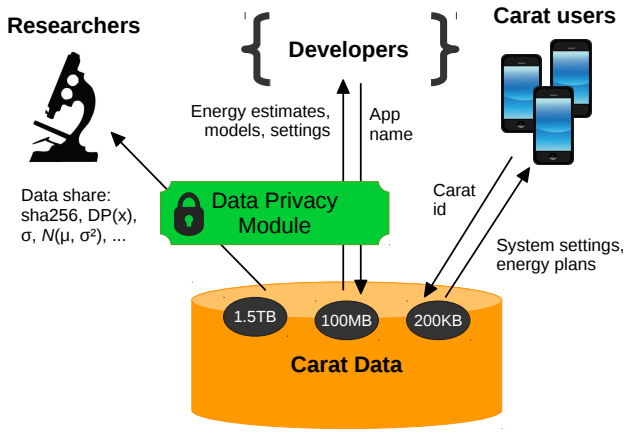

Fig. 3. Carat data sharing model for researchers, developers, and users.

IV. Data Sharing Challenges

One of the challenges of sharing large datasets is their sheer size. Multiple terabytes of data can take days to download even when the data provider has invested in the network capacity necessary. As datasets become more popular, more network resources are needed. One possibility with our dataset is sharing it through Amazon's S3 storage service. This allows sharing arbitrarily large datasets, and enables network transfer costs to be paid by the downloader so that the data provider does not receive an overwhelming bill simply because they chose to share data free of charge. However, using Amazon S3 requires both the data provider and all interested parties to have Amazon accounts, which locks the data behind a single cloud vendor.

Raw data collected in the wild can contain errors, such as failure to obtain a measurement, erroneous measurement or missing values, and meaningless defaults. Errors may even be used to identify individual devices [7]. Effective data cleaning procedures should be applied to any data before publication.

Sharing the privacy-sensitive data can cause a multiple problems, for example, users might be identified from the data against their will, their location might be discovered, and application ideas under development revealed prematurely. Therefore, sharing raw data includes risks for both users participating in the data collection and the authors entrusted with managing the data. Hashing user identifiers, or application names may help in certain situations but also limits the value of the information and precludes some high utility use cases.

Most shared datasets come with licenses and agreements how they can be used and in which conditions. Even with proper licensing policy, any opportunities for misuses should be kept to a minimum. If these challenges preclude sharing the raw data, we plan to open access to the results of the analysis as well as statistics found in the data.

V. Planned Solutions

For the Carat users, our application presents a simple statistical analysis of their devices performance compared to the others (so called J-Score). Some statistics are also provided from the whole community, for example, distribution of well-behaved and energy-hungry applications, and popular device models. In the Carat website, we also provide a view into the top two hundred most energy-intensive applications on both iOS and Android, searchable by date. As further data sharing towards the users of Carat, we are planning to add system settings recommendations, and "energy planning", allowing users to create energy management plans for a day, a week, or a longer period of time.

For developers who want to utilize the battery information, we can make available an API providing an ever-growing set of statistical values. Thus, without sharing the raw data or user identifiers we can offer a view into real-time energy consumption in the Carat community. Based on a search by the application process name, the API could return, for example:

- Number of users without specifying user identifiers
- Energy estimate as the expected average consumption %/s, with estimated error
- Energy estimates and corresponding error estimates in certain system setting combinations, if enough data is available, for example with different screen brightness levels (automatic, or manual between 0 to 255), network type (Wi-Fi or mobile network), Wi-Fi signal strength (one to four bars), CPU usage (high, medium, or low), and battery temperature.
- Distribution of models and operating systems
- Energy estimates of different models and operating systems, if enough data is available (e.g. 100 users per model)
- Package signing identities, for malware detection purposes (whether the developers recognize their key)

Fig. 4 gives an example of the developer API. Firstly, the developer downloads the Carat SDK and signs it by the same developer key they use for the applications they want to monitor. If the key matches to ones seen in the Carat data, access is granted. Secondly, the developer sets the criteria for the query and gets back the application's average energy consumption where the condition holds, e.g. screen brightness is 255. The tasks of the data privacy module are to control access of the developers, protect the privacy of individual users and applications with very small user bases, and to compute the anonymized, differentially private data for research purposes.

For researcher community, the benefit of datasets are many. We can use datasets from other researchers to obtain statistical facts and to verify our findings with reference data. To obtain the utility of a dataset while maintaining user privacy, we can remove or anonymize privacy sensitive elements of the data. Firstly, any kind of user or device identifier uniquely identifies an individual inside the data, even if it cannot be associated to personal information such as an email or phone number. Using the device model or the application as an identifier can be sufficient in some cases. However, it may be necessary to tell apart the devices in the data.

Some previous mobile application data sets remove or hash also application package names to protect the privacy of app developers. If the most popular apps of the community are known, such hashes can be decoded. To mitigate this, we can remove

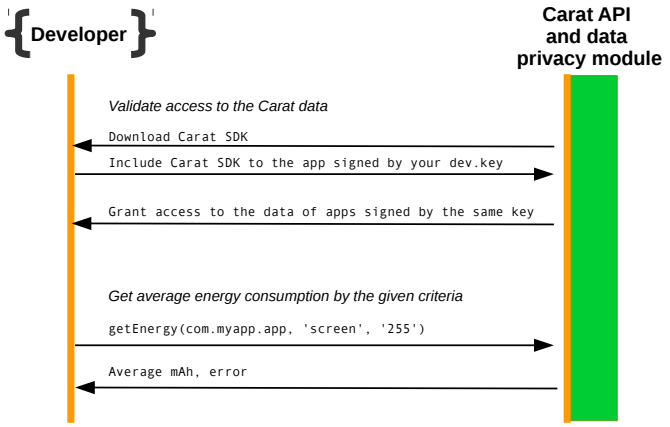

Fig. 4. Example of the Carat developer API.

applications with a small number of users from the dataset, so that applications in internal and external development will not be exposed to the public prematurely. Another way to anonymize application information is to represent applications with a sequence number and the functionality they provide, e.g., as a set of keywords instead of application names, such as Google Play category names.

In addition to modifying the raw data by anonymizing it and restricting the fields shared, we consider how to share only distributions or sufficient statistics in a differentially private manner. This method allows to share data in the form of statistical distributions, allowing high reproducibility of results for the statistical tests that are supported at the moment of publication. However, some tests may not be possible with this limited, differentially private set of statistical indicators.

Based on statistical distributions, we can generate artificial device data, which can then be shared without privacy implications. This kind of synthetic dataset can utilize techniques such as k-anonymity and l-diversity. However, hidden effects not represented by the statistical distribution can then be missed. Further work is required to determine the optimal way to share large-scale datasets with the research community.

VI. Summary and Discussion

This paper has discussed the challenges and possible solutions of sharing a large-scale mobile analytics dataset containing several fields that can be used to uniquely identify a device. Such fields include (random hash) user identifiers, applications used by users, and even the battery level and Wi-Fi signal strength. To allow sharing such datasets, we propose using an API that implements differential privacy, limits developer access to data from their own applications, and suitably anonymizes data for research purposes.

Datasets are used in the research community to support conclusions with statistical facts, validate results with a reference dataset, and to develop new methodologies. Data privacy techniques such as differential privacy allow sharing predetermined statistics and life-like artificial data that exhibits the same statistics as the raw data. However, applying these techniques always limits the depth of the data shared, and further work with interested parties is required to enable sharing the Carat data with maximum utility while maintaining the privacy of users and application developers.

Acknowledgments: The research reported in this article was supported in part by the Academy of Finland grants 303815 and 277498. The publication only reflects the authors' views.

References

[1] Jagdish Prasad Achara, Gergely Acs, and Claude Castelluccia. 2015. On the Unicity of Smartphone Applications. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society (WPES '15)*. ACM, New York, NY, USA, 27–36.

[2] Kumaripaba Athukorala, Eemil Lagerspetz, Maria von Kügelgen, Antti Jylhä, Adam J. Oliner, Sasu Tarkoma, and Giulio Jacucci. 2014. How Carat Affects User Behavior: Implications for Mobile Battery Awareness Applications. In *CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA, 1029–1038.

[3] Cynthia Dwork. 2006. Differential Privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II (ICALP'06)*. Springer-Verlag, Berlin, Heidelberg, 1–12.

[4] Brian Ferris, Dieter Fox, and Neil D Lawrence. 2007. WiFi-SLAM Using Gaussian Process Latent Variable Models.. In *IJCAI*, Vol. 7. 2480–2485.

[5] Arik Friedman and Assaf Schuster. 2010. Data Mining with Differential Privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*. ACM, New York, NY, USA, 493–502.

[6] Eemil Lagerspetz, Hien Thi Thu Truong, Sasu Tarkoma, and N. Asokan. 2014. MDoctor: A Mobile Malware Prognosis Application. In *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, fake, 201–206.

[7] Łukasz Olejnik, Gunes Acar, Claude Castelluccia, and Claudia Diaz. 2015. The leaking battery. In *International Workshop on Data Privacy Management*. Springer, Springer, New York, NY, USA, 254–263.

[8] Adam J. Oliner, Anand P. Iyer, Ion Stoica, Eemil Lagerspetz, and Sasu Tarkoma. 2013. Carat: Collaborative Energy Diagnosis for Mobile Devices. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 10:1–10:14.

[9] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. 2015a. Energy Modeling of System Settings: A Crowdsourced Approach. In *IEEE International Conference on Pervasive Computing and Communications (PerCom'15)*. St. Louis, MO, USA.

[10] Ella Peltonen, Eemil Lagerspetz, Petteri Nurmi, and Sasu Tarkoma. 2015b. Energy Modeling of System Settings: A Crowdsourced Approach. In *Proceedings of the 13th International Conference on Pervasive Computing and Communications (PerCom)*. IEEE.

[11] E. Peltonen, E. Lagerspetz, P. Nurmi, and S. Tarkoma. 2016. Constella: Recommending System Settings the Crowdsourced Way. *Pervasive and Mobile Computing* 26 (2016), 71–90.

[12] Adam Smith. 2011. Privacy-preserving Statistical Estimation with Optimal Convergence Rates. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing (STOC '11)*. ACM, New York, NY, USA, 813–822.

[13] Hien Thi Thu Truong, Eemil Lagerspetz, Petteri Nurmi, Adam J. Oliner, Sasu Tarkoma, N. Asokan, and Sourav Bhattacharya. 2014. The Company You Keep: Mobile Malware Infection Rates and Inexpensive Risk Indicators. In *23nd International World Wide Web Conference, WWW '14, Seoul, Korea, April 7-11, 2014*.

[14] Daniel Wagner, Andrew Rice, and Alastair Beresford. 2014. Device Analyzer: Understanding Smartphone Usage. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 131. Springer International Publishing, 195–208.

[15] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2012. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In *Proceedings of NSDI '12: 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI '12)*. USENIX Association, 15–28.