Open Smartphone Data for Mobility and Utilization Analysis in Ubiquitous Environments

Jochen Streicher[1], Nico Piatkowski[2], Katharina Morik[2], and Olaf Spinczyk[1]

Department of Computer Science, [1]LS12 and [2]LS8
TU Dortmund University
D-44227 Dortmund, Germany
{jochen.streicher,nico.piatkowski,
katharina.morik,olaf.spinczyk}@tu-dortmund.de
http://{ess,www-ai}.cs.uni-dortmund.de

Abstract. The development and evaluation of new data mining methods for ubiquitous environments and systems requires real data that were collected from real users. In this work, we present an open smartphone utilization and mobility dataset that was generated with several devices and participants during a 4-month study. A particularity of this dataset is the inclusion of low-level operating system data. Additionally to the description of the data, we also describe the process of collection and the privacy measures we applied. To demonstrate the utility of the data, we performed two example analyses, which are also presented in this paper.

1 Introduction

Today's mobile phones are able to produce a vast amount of valuable data. Produced by several physical and logical sensors, the data provides knowledge about the owner as well as his environment. Several studies have shown that smartphones can be used as an effective tool to gain insights into patterns of human behavior and interaction that were not available before. Notable examples are the datasets of the MIT Human Dynamics Lab like the Reality Mining dataset [1] or the Lausanne Data Collection Campaign [2]. The latter was however only available for participants of the 2012 Nokia data challenge. [3]

While smartphones are certainly an excellent *tool* for research, it is not less important to consider how data collection campaigns can help to improve these devices and the respective infrastructure. Previous research has shown that insights into utilization and mobility patterns of mobile devices are indeed of value for that purpose. This concerns the mobile network infrastructure [4] as well as the user experience with respect to the devices. A limiting factor to user experience is certainly the lifetime of a smartphone's battery. Much research has been conducted towards the use of user-specific mobility and utilization patterns to increase the energy-efficiency of mobile devices, like the reduction of GPS utilization via location prediction [5] or accelerated file prefetching. [6]

Research on these problems requires real data collected on real devices and from real users. While high-level data like location and phone call logs might

be sufficient for some of the problems, others also need data concerning the device, not only its owner. *Device Analyzer*[1] is an Android application collecting data from thousands of devices all over the world in order to get insights into utilization patterns, with the explicit goal to provide crucial information for the improvement of future smartphones. To guarantee complete anonymity all privacy-critical identifiers (e.g., cell tower IDs or MAC-addresses) are hashed with individual salts. This makes the dataset unavailable for the analysis of social interaction. Also, it is not clear when the data will be available.

Our proposed dataset shares commonalities with all three mentioned examples, but features all of the following: 1.) It contains operating system level data, 2.) not all identifiers are hashed with individual salts, and 3.) it is openly available at http://sfb876.tu-dortmund.de/mobidata.

The remainder of this paper is structured as follows: In Section 2 we will shortly describe how we collected the data and ensured the privacy of our participants, while Section 3 describes the resulting dataset. In Section 4 we present two exemplary analyses performed on these data. Section 5 concludes the paper.

2 The Collection Process

We started with 11 participants, who all were members of our collaborative research center. During our summer school in 2012, we additonally collected data from 11 attendees. For this purpose we used *MobiDAC*, our flexible infrastructure for data collection on Android-based smartphones. MobiDAC allows experimenters to use the participating devices like programmable sensor nodes. Operators write *sensing modules* that perform the actual data acquisition on the device. These modules may be uploaded to respective devices and remotely started or stopped. When a module is running, it is collecting, possibly preprocessing and saving data locally on the device. Data is sent back to the experimenter when certain conditions are met, like an established Wi-Fi connection. Currently, a modified version of the Scripting Layer for Android (SL4A)[2] is used to execute the sensing modules.

2.1 Modus Operandi

We used both the Android-API as well as Linux' virtual file systems (VFS) "`/proc`" and "`/sys`" as data sources. When the device was awake, most of the data was collected high-frequently (temporal resolution of two seconds) or via callbacks from Android. To reduce the amount of data that had to be transmitted from the device, we only recorded changes to data values. Every 60 seconds, we took a snapshot of all data from the virtual file systems and started sensor sampling for two seconds with the highest possible frequencies. A Bluetooth scan was started every five minutes. Every two hours, the periodically sampled

[1] Device Analyzer website: http://deviceanalyzer.cl.cam.ac.uk/
[2] SL4A can be found at: http://code.google.com/p/android-scripting

data was recorded completely (not only changes). As opposed to Symbian-based phones, which were used for some of the data collection campaigns mentioned in the introduction, Android phones try to *suspend* whenever possible. This happens, when the screen is off and no application is keeping the device awake by means of a *wake lock*. During the suspended state, no data can be collected at all. Thus, we explicitly wake the device from its sleep every 60 seconds and perform a full acquisition of all data values with the respective intervals.

2.2 Privacy Preservation

Since this version of our dataset is truly open and available to anyone, we were obliged to be especially careful in the process of ensuring privacy. We treated data in the following ways:

- Everything that uniquely identifies a participant is *globally consistently* replaced with a random value. This is also true for all identifiers from interaction with other entities (e.g., MAC-addresses and SSIDs) as well as for the names of installed and running application packages and processes.
- Mobile network cell information was replaced by *locally consistent* random values for each participant. This means that the mapping of cell identification (CID) and location area code (LAC) is different for every participant.

3 The Data

We collected data from various hardware and software subsystems, namely communication (Wi-Fi, Bluetooth and mobile), sensors, power supply, the Linux kernel and Android's application framework. This section coarsely describes the contents of the dataset resulting after the privacy-preserving measures.

3.1 Contents

The data may be categorized into high-level user context, external sensing, and system internals.

High-Level User Context is utilization data that contains direct hints to the participant's current activity and context. This includes the state of the display (on/off, brightness) and the phone (idle, ringing, or off the hook). Also the currently running packages belong to this category. Settings can also indirectly tell about the participant's context. For example, turning the phone to silent mode, when it was set to play a ringtone before, is a hint that the situation changed to one that prohibits phone noise, like a meeting or a cinema. Besides audio settings, also the communication settings, whether Bluetooth or Wi-Fi is enabled, or whether the device is in airplane mode, belong to this category.

Sensing data is obtained from various physical sensors as well as positioning and communication hardware. The physical sensors measured acceleration, magnetic field strength, orientation and light intensity. When the participant allowed it, also information about current altitude and speed were obtained from the GPS hardware. Also, communication devices can be used to sense the presence or even the signal strength of (potential) peers. Whereas Wi-Fi and baseband processors deliver information about stationary communication peers (access points and cell towers) and are thus feasible for positioning, Bluetooth delivers information about mobile communication peers.

System Internal data mainly describes the overall usage of the system's resources like the CPU, the battery, the main memory and the network interfaces. The use of Android *wakelocks* also belongs to this category. Since wakelocks are used to prevent the device from suspending, (application) bugs regarding their handling can severely increase energy consumption.

3.2 Structure

For every device, our dataset contains a stream of events. Every event is composed of a timestamp, an attribute name, and the new value for this attribute. Table 2 shows an excerpt of such a stream. For entity types with multiple instances, like Wi-Fi access points, attribute names contain a unique identifier for this resource (e.g., the BSSID for Wi-Fi). The appearance and disappearance of such entities is denoted with "1" or "0" respectively. For example, at time 1346837529394, package "gBRth" is started, whereas at 1346837579524, the access point "PSQdw" has gotten out of reach. The complete dataset consists of 250 million of these events. Figure 1 illustrates their distribution regarding event type and participant. Table 1 contains all attributes in condensed form. A detailed and exhaustive description can be found at the dataset's website.

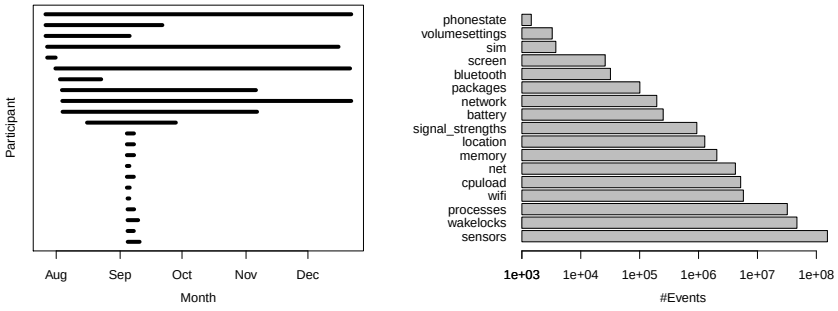

Fig. 1. Left: Period of participation for every participant. Right: Total number of events for every event type (log scale).

Table 1. Condensed names of collected attributes including the data **Cat**egory (**H**igh-Level, **S**ensing, or **I**nternal), the sampling **Int**erval, and the data **Sour**ce (Android **A**PI or Linux **V**FS). The sampling interval is either given in seconds or as the fact that we received the data as callback event (**E**) whenever it changed. Values in square brackets are placeholders for actual identifiers. An asterisk means that the according value (or identifier if in the bracket) was replaced for privacy.

Attributes	Cat.	Int.	Src.
airplanemode, phonestate	H	2,E	A
battery:{health, level, plugged, status, technology, temperature, voltage}	I	E	A
bluetooth:{, connected:[*mac_address**], device:[*mac_address**]:{, name*, class, bondstate, prev_bondstate}}	H,S		A
cpuload:{1min, 5min}	I	2,60	V
location:{gps, network}:{time, speed, altitude, accuracy}	S	60	A
media:{maxvolume, volume}	H	2	A
memory:{Buffers, Cached, Dirty, MemFree, MemTotal, Writeback}	I	60	V
net:[*device*]:{, {r, t}x_{bytes, packets, dropped, errors}, ... }	I	60	V
network:{roaming, cell:{cid*, lac*}, operatorid*}	S	2	A
notifications:vibrate, ringer:{maxvolume, silent, vibrate, volume}	H	2	A
packages:{launchable:[*package**], running:[*package**]}	H	60	A
processes:[*pid*]:{, cmdline:{*, parameters*}, state, tcomm*, {u, s, cu, cs}time, priority, nice, num_threads, start_time, vsize}	I	60	V
screen:{brightness, on, timeout}	H	2	A
self:{skip, start}			
sensors:{time, azimuth, light, pitch, roll, time, {x,y,z}force, {x,y,z}Mag}	S	120	A
signal_strenghts:{gsm_signal_strength, cdma_dbm, evdo_dbm}	S	E	A
sim:{state, operatorid*, serial*, subscriberid*}	I	300	A
wakelocks:[*name*]:{active_since, {expire, wake, }_count, last_change, {max, sleep, total}_time}	I	2	V
wifi:{, connection:{bssid*, hidden_ssid*, ip_address*, link_speed, network_id, rssi, ssid*, supplicant_state}, scan:[*bssid**]:{, capabilities, frequency, level, ssid*}}	H,S	60	A

Table 2. Example data for one device.

Timestamp	Attribute	Value	Timestamp	Attribute	Value
1346837529316	network:cell:cid	O8aal	1346837529469	cpu:load:5min	3.49
1346837529346	wifi	True	1346837529512	network:operatorid	obTLz
1346837529366	wifi:connection:ssid	WfQ4k	1346837529633	net:wlan0:tx_bytes	31342252
1346837529394	wifi:scan:PSQdw:ssid	ZvAet	1346837530254	packages:running:gBRth	1
1346837529428	ringer:vibrate	True	1346837530317	sim:serial	FUxuY
1346837529451	screen:on	False	1346837530351	phone:state	idle
1346837529454	screen:brightness	100	1346837534507	bluetooth:devices:ZOchS	1
1346837529468	battery:level	39	1346837579524	wifi:scan:PSQdw	0

4 Exemplary Analysis

Many different kinds of analysis can be imagined on the dataset presented here. Among them semantic place prediction [7], network cell prediction [4], transportation mode detection [8], frequent subsequence mining as well as the generative modeling of user or hardware behavior [9], [10]. Due to the streaming nature of our dataset, the streams abstraction [11] was used for preprocessing[3]. In the following, we will explain how the data can be incorporated in a network cell prediction task [4] as well as for smartphones power modeling [9].

Network Cell Prediction. Every mobile network connection has a unique network cell identifier. A-priori knowledge about cells that a user will visit in near future can deliver an indicator about upcoming changes in network routing and load. The network operator can automatically and pro-actively react to these changes, for example, by reserving capacity in the predicted cell. For a given set of Information I_t at time t, the network cell prediction task is to predict the next network cell that the corresponding user will visit. Forward feature selection shows, that the most important source of information is knowledge about the last k cells. For each user, one task specific dataset is generated. Therefore, the sequence of visited cells is extracted for each user and a sliding window of 10 minutes width is applied to this sequence. If the cell identifier changed multiple times within a single 10 minute window, the most frequent CID is selected. Each k consecutive windows are then concatenated to form a data row $(\text{cell}_0, \text{cell}_{-1}, \text{cell}_{-2}, \text{cell}_{-3})$, with label cell_0 and attributes cell_{-1} to cell_{-3}. Here, two windows are considered as consecutive if their time stamps do not differ more than 10 minutes. Applying a simple Naive Bayes Classifier for this problem yields only $\approx 30\%$ accuracy on the just described data. To enhance the performance for this task, cells that are only visited once can be removed from the dataset and more sophisticated methods like Support Vector Machines or Markov Random Fields can be applied [4].

Energy Modeling. Researchers have proposed a number of power models for ubiquitous systems [9], [10], [12]. Usually, power models are derived manually by using a power meter attached to one specific system instance. As a result of the model derivation process, the generated power model is at best accurate for one type of embedded system and at worst accurate only for the specific ubiquitous system instance for which it was built. It would require great effort and time to manually generate power models for the wide range of phones now available.

We now show how a simple linear regression power model can be estimated with our dataset, whereby we follow the approach that was presented by Zhang et al. [9]. The event stream is converted to a set of consecutive windows as described above. Since the energy consumption should be predicted, we consider the change of battery level as label, i.e. $y = \text{batLevel}_t - \text{batLevel}_{t-1}$. The following measurements are considered as features: mobile network and Wi-Fi signal

[3] The stream container and processors that have been written to preprocess the data for both tasks are available online at: http://sfb876.tu-dortmund.de/mobistream.

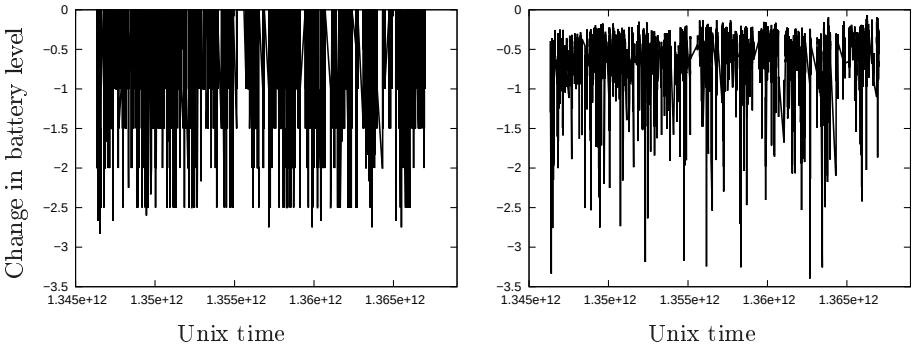

Fig. 2. Energy consumption of a smartphone as measured by the change in battery level over time. Left: measured consumption. Right: predicted energy consumption.

strength, Wi-Fi speed, number of outgoing/incoming Wi-Fi and mobile network packets in the last window, display brightness, GPS usage and CPU utilization. Figure 2 shows the measured energy consumption for one user over time on the left, and the corresponding prediction on the right. The 10-fold cross validated root mean squared error of the estimated linear model is 0.604 with a deviation of ± 0.02 and the absolute error is 0.525 (± 0.013).

Using such prediction models as a building block within a larger learning task can help to estimate the energy consumption of certain decisions since it can be used to assign costs to mobile network, display, CPU, Wi-Fi and GPS usage.

5 Conclusion

We presented our open smartphone utilization dataset collected within our collaborative research center and during its summer school and presented some analysis to show its utility. We believe that open datasets greatly help to evaluate and improve analysis methods like these. It is interesting to see that, the further down the software-hardware stack a data source resides, the more data is generated and the more data is actually needed to obtain meaningful results. We see this as an indication towards the need for data collection frameworks that allow for flexible preprocessing and data aggregation.

Acknowledgments

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Data Analysis", project A1. We would also like to thank our collaboration partners from the EcoSense project at Aarhus University for providing technical support.

References

1. Eagle, N., Pentland, A.: Reality mining: Sensing complex social systems. Personal and Ubiquitous Computing **10**(4) (2006) 255–268
2. Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. Proc. ICPS, Berlin (2010)
3. Laurila, J.K., Gatica-Perez, D., Aad, I., Jan Blom, Olivier Bornet, T.M.T.D., Dousse, O., Eberle, J., Miettinen, M.: The mobile data challenge: Big data for mobile computing research. In: Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
4. Michaelis, S., Piatkowski, N., Morik, K.: Predicting next network cell IDs for moving users with Discriminative and Generative Models. In: Mobile Data Challenge by Nokia Workshop in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
5. Chon, Y., Talipov, E., Shin, H., Cha, H.: Mobility prediction-based smartphone energy optimization for everyday location monitoring. In: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. SenSys '11, New York, NY, USA, ACM (2011) 82–95
6. Fricke, P., Jungermann, F., Morik, K., Piatkowski, N., Spinczyk, O., Stolpe, M., Streicher, J.: Towards Adjusting Mobile Devices To User's Behaviour. In: Analysis of Social Media and Ubiquitous Data. Volume 6904 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg (2011) 99–118
7. Huang, C.M., Jia-Chin Ying, J., Tseng, V.: Mining users' behavior and environment for semantic place prediction. In: Mobile Data Challenge by Nokia Workshop in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
8. Stenneth, L., Wolfson, O., Yu, P.S., Xu, B.: Transportation mode detection using mobile phones and GIS information. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '11, New York, NY, USA, ACM (2011) 54–63
9. Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R.P., Mao, Z.M., Yang, L.: Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. CODES/ISSS '10, New York, NY, USA, ACM (2010) 105–114
10. Dong, M., Zhong, L.: Self-constructive high-rate system energy modeling for battery-powered mobile systems. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. MobiSys '11, New York, NY, USA, ACM (2011) 335–348
11. Bockermann, C., Blom, H.: The streams framework. Technical Report 5, TU Dortmund University (12 2012)
12. Kjærgaard, M.B., Blunck, H.: Unsupervised Power Profiling for Mobile Devices. In: Proceedings of the 8th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2011), Springer (2011)