Crowdsensing-based Smartphone Use Guide for Battery Life Extension

Yohan Chon [*] yohan@cs.yonsei.ac.kr
GwangMin Lee [*] lkm1991@yonsei.ac.kr
Rhan Ha [†] rhanha@hongik.ac.kr
Hojung Cha [*] hjcha@yonsei.ac.kr
[*] Department of Computer Science, Yonsei University, Seoul, Korea, Republic of
[†] Department of Computer Engineering, Hongik University, Seoul, Korea, Republic of

ABSTRACT
With the increasing popularity of smartphones, battery life is among the most crucial issues for mobile users. This paper presents a crowdsensing-based use guide to extend the lifetime of smartphones. The system answers a question raised by phone usage: *Why is my phone battery draining quickly compared to others phones despite running the same applications?* The proposed system pinpoints the major causes of battery drain in terms of both hardware and software aspects. In relation to the hardware aspect, the system quantifies degree of battery aging as a ratio metric; an estimate of 50% indicates that the battery is at half of full capacity, meaning that battery usage time is approximately half that of a new battery. The system automatically profiles battery age based on charging duration data collected by crowdsensing. In its software aspect, the system guides phone configuration to extend application usage times. The system mines large-scale usage data to infer the major energy holes in a user's phone usage. The scheme works autonomously without user intervention and does not require any external equipment. Extensive evaluation with 3,000 users demonstrated that the proposed scheme successfully extends battery life for typical mobile users.

Author Keywords
Mobile Crowdsourcing; Smartphone Sensing

ACM Classification Keywords
H.4.m. Information Systems Application: Miscellaneous

General Terms
Design; Experimentation; Measurement; Performance

INTRODUCTION
With the widespread use of mobile devices such as smartphones, battery life has become a critical issue [6]. Mobile users extensively deploy mobile device resources for diverse applications (e.g., 3G, LTE, WiFi, Camera, GPS, or CPU), but insufficient battery capacity significantly undermines the user experience. Extensive research has been conducted to develop a wide spectrum of battery management policy [19, 24, 34] to extend the battery life of mobile devices. Traditional approaches proposed transparent policies that automatically managed battery life without user intervention. More recent work has focused on use guidance for mobile devices, based on users' behaviors [3, 9, 22, 29]. The key motivations of these works are that (1) a user can take effective action to extend battery life, and (2) the performance of user-centric management is effective for extending the lifetime of smartphones. Designing a user-centric management scheme is indeed challenging. To make user behaviors more energy-efficient, the scheme needs to transform battery-related information into more intuitive one for typical (i.e., non-expert) users, providing adequate information at an appropriate time.

Our work is motivated by a question commonly raised by phone usage: *Why is my phone battery draining quickly compared to others phones despite running the same applications?* Figure 1 shows a conceptual example. Two users use the same smartphone (e.g., Samsung Galaxy S4) and the same application (e.g., Facebook), yet user A lasts only one hour while user B lasts two hours using the same

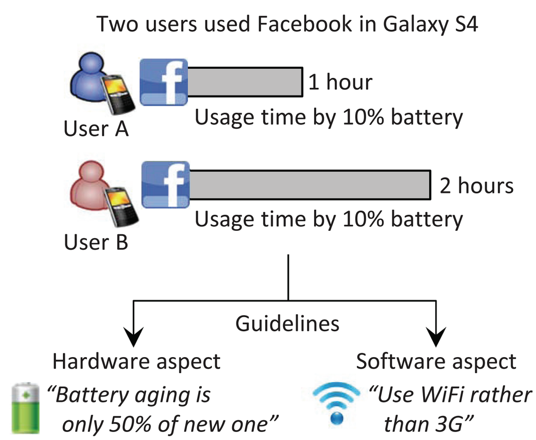

Figure 1. Conceptual example of guideline in the proposed system. Two mobile phones show different usage times despite running the same application on the same model of phone. The system provides guidelines for extending battery life in terms of both hardware and software.

UbiComp '16, September 12-16, 2016, Heidelberg, Germany
© 2016 ACM. ISBN 978-1-4503-4461-6/16/09 $15.00
DOI: http://dx.doi.org/10.1145/2971648.2971728

10% of battery capacity. What causes differences of usage time where two users were using the same application on the same hardware? Could user A extend their usage time by one hour? Our idea is that the answer is inherent in the usage behavior of user B as it relates to phone/application configuration or battery status. Based on this idea, we focus on providing appropriate guidelines to users for extending battery life of smartphone from crowd-based phone usage information.

To begin, we first analyzed smartphone usage information collected by 3,000 mobile users to establish observed differences in usage time while running the same application on the same model of phone. Based on that analysis, we designed a crowdsensing-based use guidance system for extending the smartphones battery life. The system mines large-scale mobile user usage data to investigate the relationship between usage behavior and battery life, and then provides appropriate information to each user for extending that life. The information was then translated into intuitive guides for typical users, for both hardware and software aspects. In respect of hardware, the battery aging of mobile devices is quantified as a ratio metric. For example, an estimate of 50% indicates that battery capacity is at half of full capacity, meaning that battery usage time is fundamentally half that of a new battery. In relation to the software aspect, the system provides a guideline for phone configuration to extend usage time of applications, inferring controllable features related to energy consumption from an individual's phone usage. The scheme works autonomously (without user intervention) and requires no external hardware or modification of the operating system. The contributions of this work can be summarized as follows.

• We analyzed the differences of usage time when running the same application on the same model of phone were analyzed, based on the analysis of data collected by 3,000 mobile users.

• We proposed a crowdsensing-based use guidance system that gives intuitive guidelines for both hardware and software usage to extend battery life.

• Based on quantitative and qualitative analysis, the impact of the proposed system was validated.

PRELIMINARIES AND MOTIVATION

The lifetime of a smartphone is keenly dependent on usage of various resources (e.g., 3G, LTE, WiFi, Camera, GPS, or CPU) and on battery capacity. The motivation of our work is that the battery life of smartphones can differ even where the same model of phone (i.e., same battery capacity) is running the same applications. The first step was to quantify inherent differences of battery life when the same hardware model runs the same applications. We then address the potential to extend battery lifetime in terms of both hardware and software.

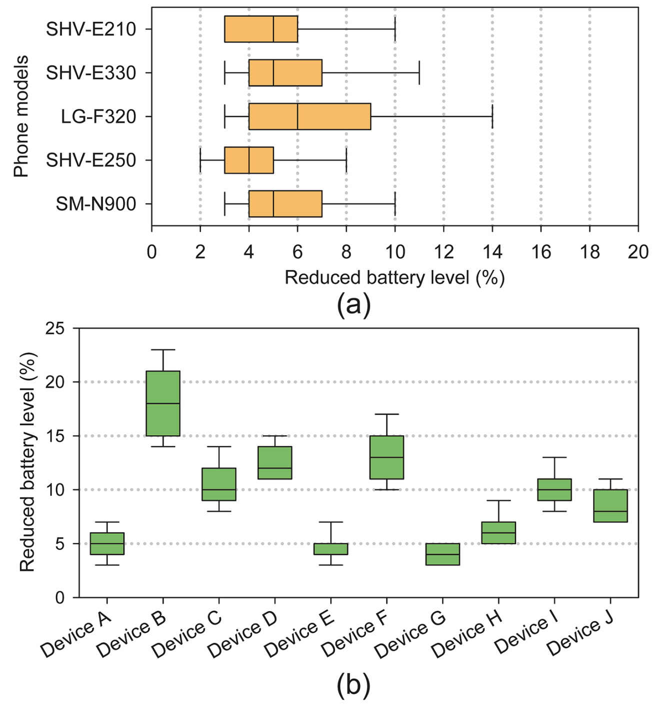

Figure 2. Reduced battery level for use of Facebook application during a 10-minute period in (a) the five most popular models and (b) a single model. Box indicates lower quartile, median, upper quartile and whisker indicates 10% and 90% of observation.

Difference of Usage Time

We explore differences in usage time when running the same applications on the same hardware model. For the analysis, we deployed a simple application to approximately 3,000 users for a period of one month in order to collect smartphone usage information. The application records battery level, charging state, current, phone state, running applications, and usage of components (i.e., WiFi, 3G/LTE, GPS, and CPU). The detail of data collection is described in Section 4.1.

Figure 2(a) shows the reduced battery level for use of the Facebook application in a 10-minute period. Usage information was collected from the five most popular models (i.e., Samsung Galaxy Note 3, Samsung Galaxy Note 2, LG Optimus G2, Samsung Galaxy S4, or Samsung Galaxy S3). The number of users for each model was 246, 230, 206, 179, and 173, respectively. To ensure fair comparison of battery levels, we employed the data collected from 30%to 70% battery level since capacity per 1%of battery level varies at low/high battery levels [12, 32]. As shown in Figure 2(a), reduced battery levels differ significantly. In the LG Optimus G2, the reduced battery level is from 2% to 18% (i.e., the difference is 16% at most). These results indicate that the usage time of applications varies significantly even though the applications were running on identical models of phone. For example, although John uses the same phone (e.g., Optimus G2) as

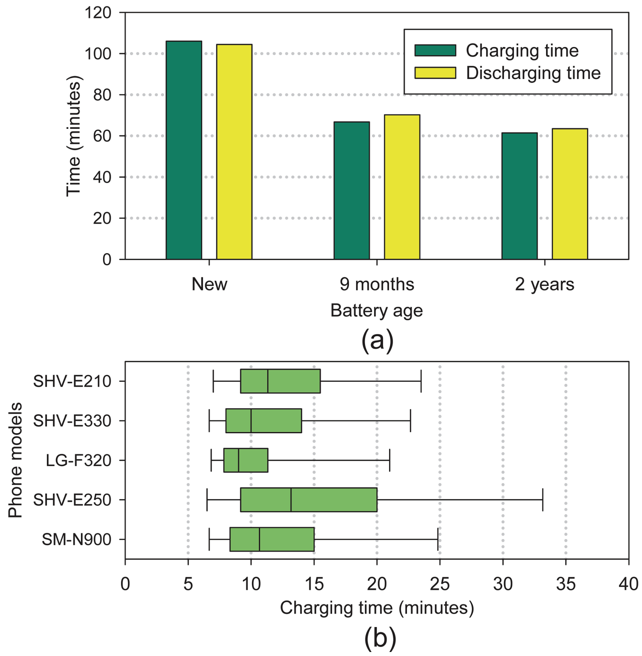

Figure 3. (a) Charging and discharging times for three batteries of different ages (new, 9 months old, and 2 years old). (b) Distribution of charging times for 10% battery levels between 30% and 70% levels in the five most popular models of phone.

his friend, the battery life of his smartphone may be significantly shorter, despite running the same applications.

The variation in usage time may be caused by differences in application usage among individual users. Figure 2(b) shows the reduced battery level for use of the Facebook application during 10 minutes on a single device. As compared to the variations among devices in Figure 2(a), the variation of battery level for each device is almost constant: i.e., the difference is only 9% at most, and 2% at least. The results show a significant observed difference in usage time between different devices but not for a single device. We further explore the causes of such differences in the following sections.

Different Battery Ages

Difference in battery capacity is a fundamental cause of difference in usage time. We observed the variation in usage time in identical phone models, which in principle have exactly the same battery capacity. However, in practice, battery capacity would vary because of battery aging effects, even for identical models. Battery aging refers to how the performance or health of a battery tends to deteriorate or diminish gradually due to irreversible physical and chemical changes that take place with usage [27]. Battery aging leads to reduced usage times for mobile devices [28].

No quantitative metric for estimating battery age in mobile devices is currently available in the literature. We infer battery capacity by estimating battery charging time, which is directly related to battery capacity. Figure 3(a) shows the charging and discharging times for three batteries in phones (Samsung Galaxy S3 GT-i9300) of different ages (i.e., new, 9 months old, and 2 years old). We measured the time between 30% and 70% battery levels and applied a steady current to the battery using external equipment. Charging and discharging times are symmetric, with only a 3.2% difference, and charging time decreases as battery age increases. The results indicate that the battery capacity can practically be estimated by measuring charging time.

Resource	Power (mW)		Note
CPU	245 MHz	194	-Lower frequency consumes less power than higher one
	614	348	
	998	554	
Cellular	532		- WiFi consumes less power than cellular
WiFi	100 packets per sec	199	
	200	263	
	500	381	
Display	1 brightness	421	- Lower brightness level consumes less power than higher one
	50	704	
	100	892	
GPS	347		

Table 1. Power consumption by resources in Nexus One. Note indicates well-known knowledge about power consumption by resource.

Figure 3(b) shows the charging time associated with a 10% change in the battery level, restricted only when the overall level was between 40 to 70% in the five most popular models of phone. To ensure fair comparison of charging time, we employed the data collected by AC charging with screen off. The median of charging time is 10.3 minutes, but charging time varies from 5.7 minutes to 37.2 minutes, according to the model of phone. For the Samsung Galaxy S3, the longest charging time is 5.1 times of the shortest one–in other words, the battery life of one user's phone could be about 20% that of another's. These results confirm that battery aging may be a main cause of reduced battery life in mobile devices. In the present case, we provide the battery aging information as a quantitative metric to mobile users.

Differences in Resource Usage

The use of various resources (e.g., 3G, LTE, WiFi, Camera, GPS, or CPU) severely affects smartphone usage time. Energy consumption by resource is well established in the literature [33]–for example, the use of WiFi consumes less energy than 3G, and lower screen brightness consumes less energy than higher brightness. Table 1 shows energy consumption for several resources. We measured the energy consumption using the Nexus One. To maintain consistency of measurement, we used the Monsoon power monitor to deliver a constant voltage of 4.0 V and halted all applications other than system service. Although the present study tested energy consumption on one specific device, energy consumption trends are widely studied in Android smartphones. In our system, we used the information about energy consumption in smartphones to mine major features of phone usage in extending battery life.

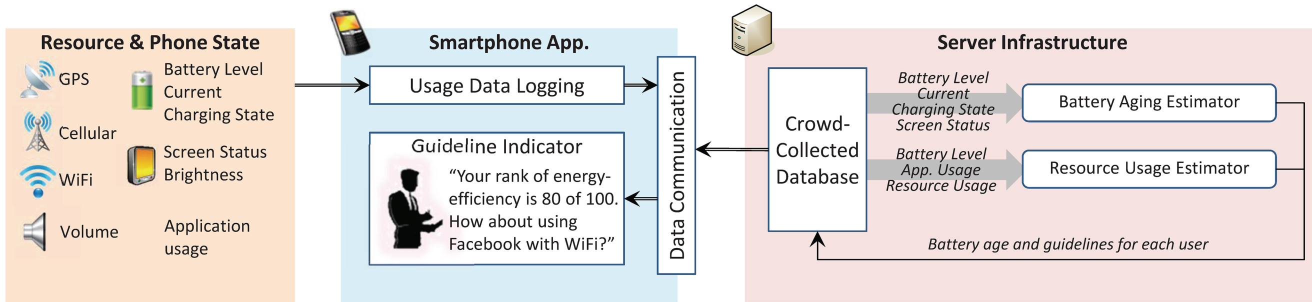

Figure 4. Overall process of the proposed system.

CROWDSENSING-BASED USE GUIDANCE FOR SMARTPHONES

The goal of use guidance is to induce change of user behaviors with a view to extending battery life of smartphones. We first describe the usage scenario of the proposed system, then present the technical details. Our system runs as a background service in mobile phones. The system transparently collects usage data and information about smartphone states such as battery status, screen status, WiFi usage, and so on. The collected data is uploaded to a server, where the data is analyzed to establish relevant guidelines for each user. Finally, the user receives the guideline as popup notification on the smartphone for extending battery life. A user can choose to perform the guideline as is, or to ignore it. In other words, the final decision about the guideline is entirely dependent on the user's intentions.

Overall Architecture

Figure 4 shows the architecture and data flow of the proposed system. The system comprises two components: (1) a smartphone app and (2) server infrastructure. The smartphone app transparently collects information about usage and phone states. All data is subsequently uploaded to a server infrastructure when WiFi is available. The server infrastructure collects, stores, and analyzes data received from the smartphone app. All uploaded data is processed by both a battery aging estimator and a resource usage estimator in an attempt to estimate battery age and major features related to energy consumption. The results are transferred to the smartphone app to generate personalized guidelines for extending battery life. Finally, the smartphone app visualizes the guidelines for each user.

Battery Age Indicator

The smartphone app includes the following components: (1) usage data logging, (2) data communication, and (3) guideline indicator.

Usage Data Logging

Usage data are primarily logged by means of an Android background service that collects data about phone states and application usage. Table 2 shows the type of usage data and the method used for collecting data. The collection of data is not periodic but is based on an event-driven method. To estimate battery age, we collect battery-related information such as charging state, capacity, remaining level, voltage,

Resource	Usage Data	Collection
Screen	Status (off, on)	Event-driven
	Brightness (1 to 100)	Event-driven
Battery	Level (1 to 100)	Event-driven
	Current (mA)	Event-driven
	Charging state (none, usb, ac)	Event-driven
	Capacity	Once
	Voltage (V)	Event-driven
WiFi	State (off, on, connected)	Event-driven
3G/LTE	State (off, on)	Event-driven
Speaker	Volume (0 to 100)	Event-driven
Bluetooth	State (off, on, connected)	Event-driven
GPS	State (off, on)	Event-driven
Application	Name (package name)	Periodic
	Background (number of processes)	Periodic

Table 2. Type of usage data and method of data collection.

and current. Current information is not available for Android, thus we collected it through sysfs in the kernel. Note that access to sysfs does not require root permission. To estimate resource usage, we collected information about application usage (i.e., name of foreground and background applications) and phone configuration (i.e., screen status, screen brightness, volume, state of WiFi, 3G/LTE, GPS, and Bluetooth).

Data Communication

To minimize the energy overhead required for data collection, the system utilizes a simple approach previously described in the literature [7]. Data communication between smartphone app and server infrastructure occurs only when (1) a WiFi connection is available, and (2) the phone is line-powered. We also limit the data communication once a day, reducing the impact on battery life and mobile bandwidth to negligible levels.

Guideline Indicator

The smartphone app notifies the user about personalized guidelines received from server infrastructure. Table 3 shows the list of guidelines for extending the battery life of mobile devices. We utilize energy-related findings from the literature, which are well known to researchers and experts but are normally unknown to typical users. The guideline is visualized as a popup window in mobile devices. The main policy of the guideline is (1) to provide a rank of energy efficiency among users of the same hardware models and (2) to report information when a user used guideline-related

Pre-messages	
Battery	Your battery age is [*x*]% which ranks [*y*]th among [*n*] [*device_name*] users.
Application	Your energy-efficiency of [*app_name*] usage is ranked [*y*]th among [*n*] [*device_name*] users.
Information about energy consumption	
Screen	Lower brightness consumes less power than higher one.
Battery	Big overhead to phone while charging accelerates battery aging.
	A [*x*]% of battery age means that your battery life is only [*x*]% of that of the new battery's.
Network	WiFi consumes less power than 3G/LTE.
	Use of network reduces battery life greatly.
Speaker	Lower volume consumes less power than higher one.
Bluetooth	Use of Bluetooth reduces battery life greatly.
GPS	Use of GPS reduces battery life greatly.
Application	Many background app reduces battery life greatly.
Post-messages	
Screen	How about lowering screen brightness?
Battery	How about using [*app_name*] without charging?
	How about using new battery?
Network	How about using [*app_name*] with WiFi?
	How about turning off 3G/LTE and WiFi?
Speaker	How about lowering volume?
Bluetooth	How about turning off Bluetooth?
GPS	How about turning off GPS?
Application	How about terminating background services: [*app_name*]?

Table 3. Usage guides used in the system. The system delivers a message combined by a pre-message, information, and a post-message.

applications or performed guideline-related behaviors. For example, when a user used heavy application such as a 3D game with battery charging, the system provides a guideline that estimates battery age, such as "*Your battery age is 40%, which ranks 70th among 100 Galaxy S3 users. The significant overhead to your mobile device while charging could accelerate battery aging. How about using the 3D game without charging?*" Similarly, when a user is using the Facebook application via 3G while a WiFi network is available, the system offers the following advice: "*Your energy-efficiency for Facebook usage is ranked 80th among 100 Galaxy S3 users. WiFi consumes less energy than 3G. How about using Facebook with WiFi?*" We expect that users could learn these guidelines and change their behaviors to extend battery life.

The pre-message is generated based on the data collected by *battery aging estimator* and the information about energy consumption. The post-messages are chosen by using information inferred by *resource usage estimator*. The system simply sorts crowd-collected data to choose pre-message. The post-message is decided by heuristic comparison of inferred usage type between individual user and crowd-collected data. For example, let the user A be classified as a heavy consumer of application X and a group of users be classified as light consumers for the same application. The system then investigates the difference of resource usage between user A and a group of light consumers in order to choose the information about energy consumption and post-messages.

Server Infrastructure

As illustrated in Figure 4, the server infrastructure comprises two components: (1) a battery aging estimator and (2) a resource usage estimator. The battery aging estimator infers the battery age of each device as a quantitative metric. The resource usage estimator investigates major features related to energy consumption by applications.

Battery Aging Estimator

We present a scheme that quantifies battery aging as a quantitative metric. A simple way of estimating battery aging is to measure the battery's current consumption, using the Battery Measurement Unit (BMU). However, the BMU update rate is typically too low, and some BMUs do not provide current information. A new method other than the use of current consumption at discharging is therefore required to estimate battery aging.

We define battery aging as the ratio of actual usage time to maximum usage time for the same hardware model. An intuitive way to estimate the maximum usage time for a specific hardware model is offline training, using a brand new battery for that specific model [18]. Such manual training is, however, neither scalable nor efficient for consideration of all commercial hardware. For that reason, we utilize crowd-collected data by our smartphone app to estimate the maximum usage time of specific hardware models. To address this issue, we considered that charging and discharging curves are symmetric for Li-ion batteries [12, 32]. The idea is that if battery capacity on charging can be estimated, this information can be directly used to estimate battery capacity upon discharging, without information about the current consumption of the actual hardware device. In this way, rather than maximum usage time, we estimate the maximum charging time of specific hardware models.

Let $B = \{d, u, b, s, h, c, t\}$ be data uploaded by a user, where d is device model, u is unique identifier of a user, b is battery level as a ratio, s is screen status (on or off), h is charging state (none, USB, or AC), c is battery current, and t is a timestamp. We pre-process the collected data to generate the charging time from battery level a to battery level b $T^{i}_{[a,b]}$ for user i by using data with charging (i.e., h = USB or AC). Here, we filter out the data under two constraints. First, we do not use data at the lowest battery level (e.g., less than 40%) or at the highest (e.g., higher than 70%) because battery capacity per 1% battery level

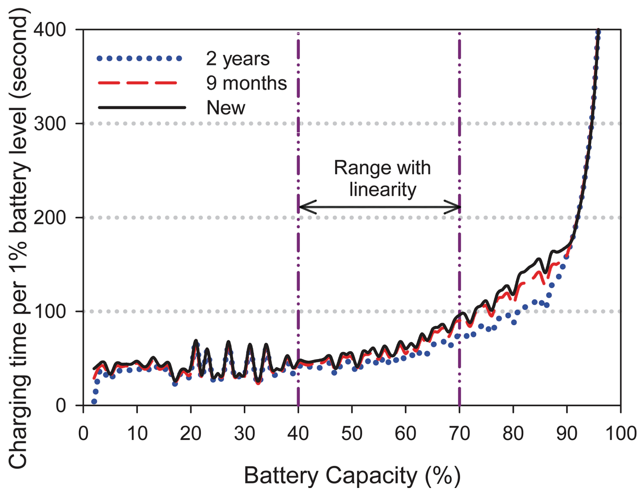

Figure 5. Trace of charging time of Li-ion battery. Charging time per 1% battery capacity shows linearity in the middle region.

significantly varies at the lowest or highest battery levels [18], we filter out data with screen on because active use of devices could increase charging time although the device is charging. We used the median value of $T^i_{[a,b]}$ to prevent bias in terms of charging time. The battery age of the user's device (ba^i) is then calculated as:

$$ba^i = \frac{T^i_{[a,b]}}{\max_{1 \le x \le n} T^x_{[a,b]}},$$

where n is the number of users who have used the same hardware model with user i's device, and a and b are parameters with $\alpha < a < b < \beta$. α and β are empirically set to filter out variation of charging time at lowest or highest battery levels.

We further process the data to generate robust estimation results. Charging current via USB (e.g., 300 mA) is lower than via AC adapter (e.g., 800 mA). Intuitively, different charging currents mean different charging times. On that basis, we normalized charging time by charging current to make a fair comparison between charging times of devices. Charging current for mobile devices is accessible online. In addition, when calculating $T^i_{[a,b]}$, we interpolate missing values between a% and b% battery level. We use the linear regression for interpolation since the charging time per 1% battery capacity shows linearity, as in Figure 5. The data were discretized into month buckets since battery age changes as usage time increases.

Finally, the system delivers the battery age of their device (ba^i) to user i. The intuitive example is that, for a certain battery, the charging time of user i's device from 50% battery level up to 60% ($T^i_{[50,60]}$) is 610 seconds with the AC adapter. Among the charging times collected by other users using the same hardware model, the maximum charging time is 1220 seconds. The battery age of user i's device, then, is $0.5 = \frac{610}{1220}$. Here, the maximum charging time among collected data is considered 100% of battery age. This crowdsensed approach enables practical estimation of battery age since the information is collected by real smartphone users and the battery charging information is rarely corrupted by users.

Resource Usage Estimator

The purpose of the resource usage estimator is to infer features that could be changed to extend the battery life of devices. Energy consumption by mobile devices always reflects phone state as well as configuration (screen brightness, use of WiFi, 3G/LTE, GPS, Bluetooth, and number of executed applications). Therefore, we extracted and categorized the features in two aspects: phone state and configuration.

The system used several features related to resources, as shown in Table 2. The discrete values are the names of foreground applications and hardware models. We used the binary features for sensor-related phone configuration such as WiFi, 3G/LTE, GPS, and Bluetooth. The system considered screen brightness, volume, and number of running services as continuous values. Based on data collected by the smartphone app, the system first groups usage data by names of foreground applications and hardware model. The system then generates the feature vectors for usage time per 1% battery capacity. Here, we discretized usage time into five bins and normalize battery capacity rather than use of raw battery level, based on estimated capacity during charging. The reason is that the actual capacity per 1% battery level varies according to remaining battery level [12, 32]. For example, despite identical overheads, a smartphone will perform a certain task for longer at 90% battery level than at 10% battery level.

After extracting features of usage information, we have four binary fPeatures and three continuous values, and so the feature vector could be 1.6 million combinations (i.e., $2^4 \times 100 \times 100 \times 10$) of values in total. Classifiers are trained using the feature vectors from collected data and iteratively find variations of feature vectors from specific users that may increase usage time. For classifiers, we used the Labeled Latent Dirichlet Allocation (L-LDA) model [30]. L-LDA is an extension of traditional LDA [4]; allowing topic models to be trained with labeled documents. The model is optimal in our case since the number of features is finite; that is, all continuous values are transformed into integer value in a finite space. Also, the reduction of battery usage can be used for label information, and the model allows multiple topics to the distribution. A separate L-LDA model is trained for each bin of usage time and can be used to find feature vectors that would be classified as longer usage time. The model generates multinomial topic distributions over vocabulary $\beta_k = (\beta_{k,1}, \cdots, \beta_{k,V})^T \sim \mathrm{Dir}(\cdot|\eta)$ for each topic k, from a Dirichlet prior η. The L-LDA model then draws a multinomial mixture distribution $\theta^{(d)}$

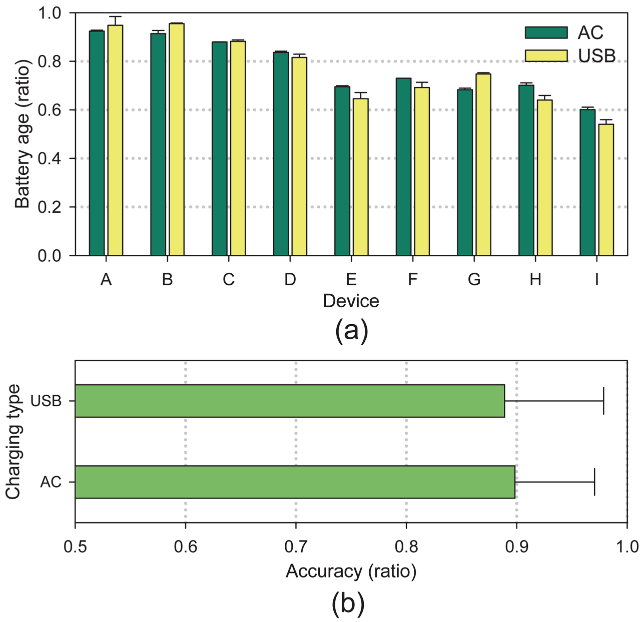

Figure 6. (a) Estimated battery age of 9 devices in the proposed scheme; (b) accuracy of battery aging estimator.

over the topics, corresponding to their labels $\theta^{(d)}$. For any vector, the final distribution $\theta^{(d)}$ will correspond to the relevance of the usage time.

EVALUATION

We evaluate the proposed system in four respects: battery aging estimator, resource usage estimator, impact on battery life, and system overhead in smartphones. We omitted the server side computation overhead because it is not critical to the user experiences. We mainly focus on the accuracy of guidelines provided by the system, which is the major objective.

Implementation and Data Collection

The proposed system consists of a smartphone client and server infrastructure. For primary smartphones, we implemented the application using Android SDK 4.0, running on commercial smartphones equipped with GSM/CDMA, WiFi, Bluetooth, and GPS. The application records battery level, charging state, battery current, phone state, running applications, and usage of components (WiFi, 3G/LTE, GPS, and CPU). The application notifies the user of guidelines in a popup window. We deployed the application through the Google Play store and collected the usage information of smartphones for a month. The dataset includes 3,626,396 samples collected from 254 hardware models of 3,121 users. The server infrastructure used the Google Cloud Server with 8 cores, 30 GB RAM compute engine and 512MB RAM, and 250GB storage database.

In the battery aging estimator, the parameters α and β were set to preserve the linearity of charging time per 1% battery level between α and β. In the device with maximum charging time, we divided the trace of charging time into three parts $[0, \alpha]$, $[\alpha, \beta]$, and $[\beta, 100]$. We then applied the

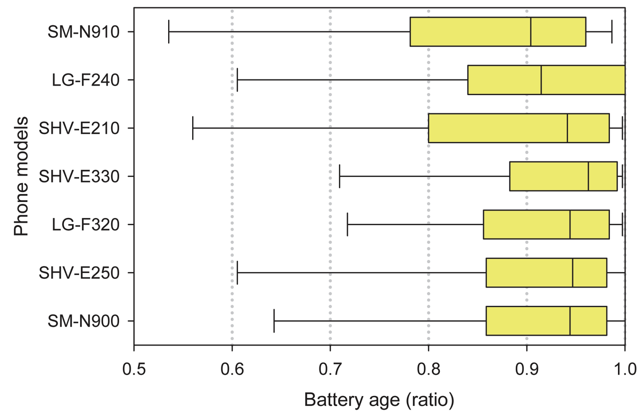

Figure 7. Distribution of battery ages in the dataset. The battery ages varied from 0.47 to 1.0.

least square regression to each part to find the linearity with minimum error e. The error in each part is denoted in least squares regression as $e^{low}, e^{middle}, e^{high}$, and we iteratively identify an $[\alpha, \beta]$ that minimizes the summation of errors $e^{low} + e^{middle} + e^{high}$. Here, we restrict the value as $\alpha <$ 50% and $\beta >$ 50% based on the characteristics of the Li-ion battery. For n samples, the complexity of the algorithm is $O(n^3)$, but computation time is trivial since n is at most 50, as in our case.

Performance of Battery Aging Estimator

We first evaluate the accuracy of the battery aging estimator. The ground truth of battery aging is, in fact, not known. Thus, we chose 9 devices from among the most popular hardware models and indirectly estimated the ground truth of battery aging. The proposed scheme outputs estimated battery age as a ratio metric. We measured the discharge time of a brand new battery three times for each device, from 100% battery level down to 0%, for the same workload. We used the discharge time of new batteries as the reference data. The error of battery age is calculated as the difference between the estimated battery age and the manually measured one, defined as follows:

$$error^i = ba^i - \frac{D^i}{D^{new}},$$

where ba^i is the estimated battery age of device i, D^i is the measured discharge time of device i, and D^{new} is the measured discharge time of the device, using a new battery.

Figure 6(a) shows the estimated battery age of the proposed scheme, charging via AC and/or USB. The age of batteries varied from 0.59 to 0.92, and the difference in ages via AC and USB is negligible (i.e., only 4.8% difference). The results indicate that the proposed system outputs robust ages in both AC and USB charging modes. Figure 6(b) shows the error of the proposed scheme. Our scheme achieves 0.89±0.07 accuracy, with a range from 0.79 to 0.98. The largest error (i.e., approximately 0.21) is observed for relatively older batteries with AC charging. The reason is that the fluctuation of charging time is relatively large

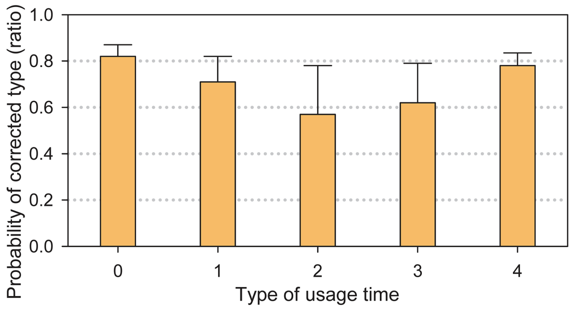

Figure 8. Accuracy of resource usage estimator. Type 0 is the shortest usage time.

with old batteries. The results show that, for typical mobile users, estimated ages are a reasonable and intuitive indicator of battery age.

We explored the relationship between the volume of data collected and accuracy of estimated battery age. Intuitively, more collected data should lead to a more accurate result; data collected for more than seven days were found to output robust results. Considering that mobile users typically charge their phone at least once a day, the proposed scheme could provide meaningful battery age information after a week.

Figure 7 shows the distribution of estimated battery ages in the dataset. We chose the seven most popular hardware models, with battery ages ranging from 0.47 to 0.98. The battery age of older models tends to less than that of newer models. For example, Samsung Galaxy Note (SM-N910, released in 2011) shows lower battery ages than ages of Samsung Galaxy S4 (SHV-E330, released in 2013). In the worst case, a specific user is using a battery with only 47% capacity as compared with a brand new battery. Among 3,121 users, 7% of users were using an old battery with less than 50% capacity. The results indicate that the proposed scheme could provide useful information to induce replacement of old batteries by new ones, which consequentially extends the battery life of devices.

Performance of Resource Usage Estimator

We investigated the accuracy of the resource usage estimator. High accuracy would indicate that the component correctly infers the relationships between used resources and energy consumption in mobile devices. We used five-fold cross-validation to evaluate the performance of the resource usage estimator. We chose three of the most popular applications: Facebook, KakaoTalk, and Chrome.

Figure 8 shows overall accuracy in classifying usage time according to feature vectors. The proposed scheme outputs 0.71 ± 0.11 accuracy across these five types of usage time. The shortest/longest usage time showed the highest accuracy (i.e., 0.81) due to the strong characteristics of the feature vectors. Otherwise, the lowest accuracy was observed in the medium range of usage time. We suspect that the error is caused by similar characteristics with other types and differences in users' interactions with applications. For example, if a user frequently touches the device, the device will consume more energy even though the feature vectors are identical. This limitation is discussed in Section 5.

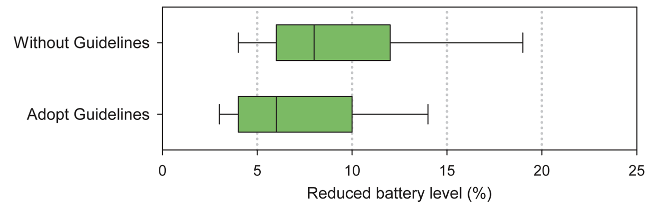

Figure 9. Effect of adopting guides on reduced battery level for using applications during same durations.

Type	3G/ LTE	WiFi	GPS	BT	Brght.	Vol.	# of Service
0	1	0	1	1	95	80	7
1	1	0	0	0	80	60	6
2	0	1	0	1	50	50	4
3	0	1	0	0	40	70	3
4	0	0	0	0	5	40	3

Table 4. Representative feature vectors for each type of usage time. Type 0 is the shortest usage time.

Table 4 shows the representative feature vectors for each type of usage time. Intuitively, greater use of resources derives shorter usage time and vice versa. The use of 3G/LTE, GPS, and Bluetooth with 95 screen brightness (*type 0*) consumes significant energy while the use of 5 screen brightness without network modules (*type 4*) consumes relatively trivial amounts of energy. We found that major applications in *type 0* are navigations while *type 4* contains audio/video players. These results are supported by the findings about energy consumption in mobile devices in the literature [15, 33]. The results show that the proposed scheme can infer energy consumption of mobile devices indirectly by using resource usage, although the exact amount of energy consumption remains unclear.

Impact of Guidelines on Battery Life

We evaluated the effect of guidelines on battery life. Based on the inferred feature vectors, the system provides smartphone usage guides for extending battery life. Guides are listed in order of frequency: i.e., WiFi-related, battery related, and screen-related guides. Volume-related and number of services-related guides are rarely delivered. The reason is that these features are not effective in reducing usage time according to the resource usage estimator.

Reduced battery level for 10 minutes usage of applications decreased by 2.1% on average when users followed the guides, as shown in Figure 9. The difference between default and changed battery level is statistically significant according to the *t*-test: i.e., $t(359) = -4.94, p < .05$. The results indicate that the provided guides are effective in extending the battery life of devices in practice.

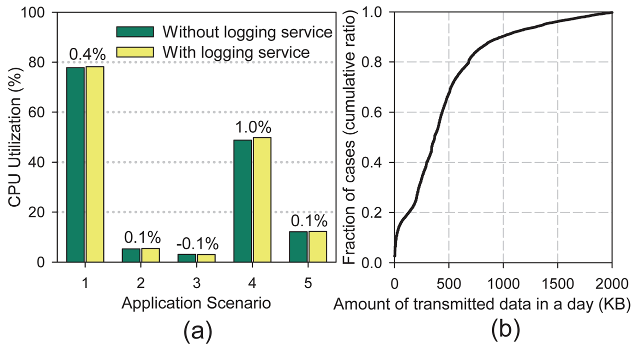

Figure 10. (a) CPU overhead of our service for data logging; (b) amount of transmitted data in a day.

Overhead in Smartphone Client

The smartphone client overheads are the computation overhead for data logging and energy consumption for transmission of collected data to the server. The major overheads incurred by data logging to store running applications and phone configurations. Figure 10(a) shows CPU utilization in the default state and in running our logging services. The additional overhead incurred by our service is trivial: i.e., only 0.3% of CPU utilization. The average amount of data transmitted in a day is 454.5 kilobytes (KB). The user sent less than 670 KB data in 80% of days, as shown in Figure 10(b). These results demonstrate that the smartphone app overhead is trivial in terms of both CPU overhead and data transmissions.

RELATED WORK

In this section, we describe prior work while highlighting the original contributions of our work.

Use guidance. Recent work has focused on user-centric guidance systems to encourage change in users' behavior toward energy-efficient usage. Rahmati et al. [29] investigated the interactions between users and batteries and claimed that a well-designed interface could guide users to change their actions and so improve energy efficiency. Most works [9, 14, 23, 31] provided the remaining battery life of mobile device to suggest termination of activated applications. Oliner et al. [22] also suggested a method for identifying problematic applications from an energy perspective by means of crowdsourcing [1]. Similarly, Jung et al. [14] provided information about energy consumption of applications in terms of usage time per 1% of battery. They claimed that users actively learned the energy efficiency of each application and changed their behaviors to extend the battery life of mobile devices. By comparison with previous works, the proposed system is unique in that the guidance system provides intuitive and direct guidelines (e.g., use WiFi rather than 3G) to typical users. We deliver well-known knowledge about energy consumption to typical users at appropriate moments.

Battery Aging. Battery aging is a traditional research issue in the chemical engineering community [5]. However, the quantitative metric for estimating battery aging in mobile devices is challenging because the modeling of chemical characteristics of Li-ion batteries is complicated. Several works [11, 20] have been conducted to understand the chemical characteristics of Li-ion batteries, but it is hard to apply this to batteries in mobile devices. Kim et al. [17] founded that batteries in mobile devices consume more power than required one, and the difference in power consumption is larger in aged batteries. This observation indicates that battery aging should be considered seriously for mobile devices. Lee et al. [18] proposed an empirical scheme to estimate battery age as a quantitative metric, but the scheme requires offline training for each battery model. To the best of our knowledge, we are the first to propose an autonomous scheme for estimating battery aging in mobile devices.

Estimation of Power Consumption. Active efforts have been made to improve the estimation of power consumption in mobile devices. Works in [19, 24, 33] addressed energy measurement for entire mobile devices, using an external gauging tool such as Monsoon or state-of-discharge (SOD) information. Dong et al. [8] proposed an online model using a battery management unit (BMU) rather than external equipment. Instead of assessing the overall device, energy consumption is monitored per hardware component and per process [24, 33], or per line of code [19]. In contrast, our system focuses on the relative comparison of energy consumption, rather than on absolute measurement, to infer energy inefficiency.

The monitoring of energy inefficiency has been extensively addressed in the literature. Works in [16, 25, 26] proposed a scheme to detect energy bugs associated with incorrect usage of wakelock. Jindal et al. [13] explored the energy inefficiency arising in suspend mode, depending on the behavior of the device driver. Abnormal energy consumption by applications and threads is considered in [2, 21]. Our system, on the other hand, detects energy inefficiency in the devices of specific users, where the same hardware models run the same application. We compared usage data across multiple devices and deliver energy efficient way by changing phone configuration. The approach most similar to our system is MPower [10], which provides the remaining battery life of mobile devices according to phone configuration. Our system, however, provides personalized guidelines in relation to phone configuration, whereas MPower shows diverse configurations without personalization.

LIMITATIONS AND DISCUSSIONS

In what follows, we describe the limitations of the present work along with future research directions. Despite the scale of our study, the participant population is not particularly diverse. Our data was collected in a single city in a single country. Thus, user behavior with respect to both energy consumption and privacy would be sensitive to characteristics such as culture and occupation. In that light, some results therefore remain to be verified by performing similar experiments elsewhere.

Context in Application Usage. The proposed system took phone configuration as accounting for the major features of energy consumption by applications. However, individual users' interactions are also an important factor in this regard. For example, although two users may have used the Facebook application with exactly the same phone configuration, user A may consume more energy than user B if user A tends to watch videos in Facebook while user B is more inclined to read texts in the posts. More frequent interaction (e.g., frequent screen touches or frequent transitions between applications) may also consume more energy. Such contextual factors in application usage are not considered in the current work. We only focused on phone configuration without considering the context of users' interactions. The design of an advanced usage model that encompasses user interaction forms part of our intended future work.

Privacy Concerns. The proposed system has a direct bearing on privacy concerns, as the system collects usage data from smartphones. In particular, collection of application usage may be a sensitive issue in terms of users' privacy. Although anonymization of application names could be applied, such a scheme fails to capture the context of applications and may therefore supply inappropriate guidelines to users. For example, "turning off the network module" is not feasible for Internet-related applications, and a user would not happy with "minimize the brightness of the screen" when he/she is using a video player application. Clearly, then, an intelligent scheme is therefore required to preserve both privacy and accuracy of the guidance system.

Maximum Values in Crowd-collected Data. The battery aging estimator considered the maximum charging time in crowd-collected data as the maximum charging time among specific hardware models. In other words, we considered the local maxima in our dataset as global maxima for the entire set. It follows that battery age results would be biased if the battery age of participants' devices was not diversely distributed. This limitation could be resolved by using sufficiently large-scale data, collected over a long period, or one-time learning for a brand new battery for each hardware mode.

Limited Analysis on Batter Level. Our system considers smartphone usages at the middle range of battery level, i.e., approximately 40% to 70% of battery level, as the labeled information. We consider the proposed approach is feasible because such middle range of battery level is normally used in daily usage pattern of smartphones. However, for more thorough analysis alternative approaches would be necessary to understand the impact of resource usage on battery levels outside this range.

CONCLUSION
In this paper, we have proposed a crowdsensing-based use guidance system to extend the battery life of smartphones. To the best of our knowledge, we are the first to implement a practical system that provides intuitive guidelines in respect of both hardware and software. We designed autonomous estimation of battery age by means of a quantitative metric. The system encourages change in phone usage toward energy-efficient behavior based on comparisons with crowd usage data. The experiments show that our approach correctly estimated battery age and key features for extending battery life. We found that users changed their behaviors to improve energy efficiency for extending battery life of smartphone. The results demonstrated that user-involved approach is practically effective in battery management. We believe that our approach can be used as a building block in guiding more energy-efficient usage of smartphones. Our future work includes the design of an advanced usage model that takes account of the context of users' interactions with applications.

ACKNOWLEDGEMENTS
This work was supported by a grant from the National Research Foundation of Korea (NRF), funded by the Korean government, Ministry of Education, Science and Technology under Grant (NRF-2014R1A2A1A11049979).

REFERENCES
1. 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 *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (CHI '14), 1029-1038.
2. Tanzirul Azim and Iulian Neamtiu, 2013. Targeted and depth-first exploration for systematic testing of android apps. In *ACM SIGPLAN Notices* , 641-660.
3. Nilanjan Banerjee, Ahmad Rahmati, Mark Corner, Sami Rollins, and Lin Zhong. 2007. Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. *UbiComp 2007: Ubiquitous Computing*: 217-234.
4. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3: 993-1022.
5. M. Broussely, Ph. Biensan, F. Bonhomme, Ph. Blanchard, S. Herreyre, K. Nechev, and RJ. Staniewicz. 2005. Main aging mechanisms in Li ion batteries. *Journal of power sources* 146, 1: 90-96.
6. Aaron Carroll and Gernot Heiser, 2013. The systems hacker's guide to the galaxy energy usage in a modern smartphone. In *Proceedings of the 4th Asia-Pacific workshop on systems* (APSys '13), 5.
7. Yohan Chon, Nicholas D. Lane, Fan Li, Hojung Cha, and Feng Zhao, 2012. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (UbiComp '12), 481-490.

8. Mian Dong and Lin Zhong, 2011. 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), 335-348.
9. Denzil Ferreira, Eija Ferreira, Jorge Goncalves, Vassilis Kostakos, and Anind K. Dey, 2013. Revisiting human-battery interaction with an interactive battery interface. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (UbiComp '13), 563-572.
10. Matteo Ferroni, Andrea Cazzola, Domenico Matteo, Alessandro A. Nacci, Donatella Sciuto, and Marco D. Santambrogio, 2013. MPower: gain back your android battery life! In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (UbiComp '13), 171-174.
11. Lijun Gao, Shengyi Liu, and Roger A. Dougal. 2002. Dynamic lithium-ion battery model for system simulation. *IEEE Transactions on Components and Packaging Technologies* 25, 3: 495-505.
12. Texas Instruments. Characteristics of rechargeable batteries, Retrieved June 11, 2016 from http://www.ti.com/lit/an/snva533/snva533.pdf
13. Abhilash Jindal, Abhinav Pathak, Y. Charlie Hu, and Samuel Midkiff, 2013. Hypnos: understanding and treating sleep conflicts in smartphones. In *Proceedings of the 8th ACM European Conference on Computer Systems* (EuroSys '13), 253-266.
14. Wonwoo Jung, Yohan Chon, Dongwon Kim, and Hojung Cha, 2014. Powerlet: an active battery interface for smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '14), 45-56.
15. Wonwoo Jung, Chulkoo Kang, Chanmin Yoon, Donwon Kim, and Hojung Cha, 2012. DevScope: a nonintrusive and online power analysis tool for smartphone hardware components. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis* (CODES/ISSS '12), 353-362.
16. Kwanghwan Kim and Hojung Cha, 2013. WakeScope: Runtime WakeLock anomaly management scheme for Android platform. In *Proceedings of the Eleventh ACM International Conference on Embedded Software* (EMSOFT '13), 27.
17. Minyong Kim, Younggeun Kim, Sungwoo Chung, and Cheolhong Kim. 2014. Measuring variance between Smartphone energy consumption and battery life. *Computer*, 7: 59-65.
18. Jaeseong Lee, Yohan Chon, and Hojung Cha, 2015. Evaluating battery aging on mobile devices. In *Proceedings of the 52nd Annual Design Automation Conference* (DAC '15), 135.
19. Ding Li, Shuai Hao, William G. J. Halfond, and Ramesh Govindan, 2013. Calculating source line level energy information for android applications. In *Proceedings of the 2013 International Symposium on Software Testing and Analysis* (ISSTA 2013), 78-89.
20. Hong Li, Zhaoxiang Wang, Liquan Chen, and Xuejie Huang. 2009. Research on Advanced Materials for Li-ion Batteries. *Advanced Materials* 21, 45: 4593-4607.
21. Xiao Ma, Peng Huang, Xinxin Jin, Pei Wang, Soyeon Park, Dongcai Shen, Yuanyuan Zhou, Lawrence K. Saul, and Geoffrey M. Voelker, 2013. eDoctor: automatically diagnosing abnormal battery drain issues on smartphones. In *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation* (NSDI '13), 57-70.
22. Marcelo Martins and Rodrigo Fonseca, 2013. Application modes: a narrow interface for end-user power management in mobile devices. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications* (HotMobile '13), 5.
23. Adam J. Oliner, Anand Iyer, Eemil Lagerspetz, Sasu Tarkoma, and Ion Stoica, 2012. Collaborative energy debugging for mobile devices. In *Proceedings of the Eighth USENIX conference on Hot Topics in System Dependability* (HotDep '12), 6-6.
24. Abhinav Pathak, Y. Charlie Hu, and Ming Zhang, 2012. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In *Proceedings of the 7th ACM european conference on Computer Systems* (EuroSys '12), 29-42.
25. Abhinav Pathak, Y. Charlie Hu, and Ming Zhang, 2011. Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices, 5.
26. Abhinav Pathak, Abhilash Jindal, Y. Charlie Hu, and Samuel P. Midkiff, 2012. What is keeping my phone awake?: characterizing and detecting no-sleep energy bugs in smartphone apps. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (MobySis '12), 267-280.
27. Valer Pop, Henk J. Bergveld, Dmitry Danilov, Paul P. L. Regtien, and Peter H. L. Notten. 2008. Battery aging process. *Battery Management Systems: Accurate State-of-Charge Indication for Battery-Powered Applications* Springer Netherlands, 111-143.
28. Valer Pop, Henk J. Bergveld, Dmitry Danilov, Paul P. L. Regtien, and Peter H. L. Notten. 2008. *Battery management systems: Accurate state-of-charge*

indication for battery-powered applications. Springer Netherlands.

29. Ahmad Rahmati, Angela Qian, and Lin Zhong, 2007. Understanding human-battery interaction on mobile phones. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services* (MobileHCI '07), 265-272.
30. Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning, 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1* (EMNLP '09), 248-256.
31. Khai N. Truong, Julie A. Kientz, Timothy Sohn, Alyssa Rosenzweig, Amanda Fonville, and Tim Smith, 2010. The design and evaluation of a task-centered battery interface. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (UbiComp '10), 341-350.
32. Min Yang and Junbo Hou. 2012. Membranes in lithium ion batteries. *Membranes* 2, 3: 367-383.
33. Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha, 2012. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In *Presented as part of the 2012 USENIX Annual Technical Conference* (USENIX ATC '12), 387-400.
34. Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing M. Mao, and Lei Yang, 2010. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis* CODES/ISSS '10, 105-114.