IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 22, 2017, accepted July 4, 2017, date of publication July 18, 2017, date of current version August 8, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2728620

Understanding Application-Battery Interactions on Smartphones: A Large-Scale Empirical Study

YAO GUO, (Member, IEEE), CHENGKE WANG, AND XIANGQUN CHEN
Key Laboratory of High-Confidence Software Technologies (MOE), School of EECS, Peking University, Beijing 100871, China

Corresponding author: Yao Guo (yaoguo@sei.pku.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2016YFB1000105 and in part by the National Natural Science Foundation of China under Grant 61421091.

ABSTRACT Current generation of smartphones is running more and more complex applications that reduce the battery life to as short as several hours. Thus, it becomes very important to understand the diversities of applications installed on smartphones and how batteries are consumed across different applications. This paper presents a large-scale battery study on smartphones focusing on diversities in applications and users. Based on application and battery traces collected on over 80 000 Android smartphones for a four-week period, we analyze the battery discharging patterns, the types of mobile applications, and the usage and energy consumption patterns for these applications. During the analysis, we introduce a novel method to calculate the energy consumption rate for each application based on coarse-grained battery data collected with a lightweight monitoring tool. Based on the results, we present some observations and discuss possible improvements on smartphone designs and mobile application development. We also compare our results to some previous studies wherever it is possible.

INDEX TERMS Mobile computing, smartphones, battery, energy consumption, user study.

I. INTRODUCTION

Smartphones have become more and more popular since the introduction of iPhone and Android-based devices. Compared to traditional feature phones, smartphones have more powerful functions and are capable of performing complex computations, while equipped with various sensors such as cameras and GPS. More and more complex mobile applications (or simply *apps*) are running on smartphones, which could reduce the battery life to as short as several hours. A recent survey shows that battery life is the single main gripe of today's mobile phone users [1].

The behaviors of smartphone users also differ greatly from traditional feature phone users. When choosing to purchase a specific smartphone, the main concerns of many users are not focused on the quality of voice calling or the length of standby time, instead of CPU processing speed and screen sizes. With the apps running on smartphones more and more complex, it will be interesting to see how much time smartphone users are relying on their phones to play games or access the Internet, compared to the time smartphones are used for traditional functions such as voice calling or sending messages.

In order to understand how smartphone batteries are consumed, many researchers have performed studies on the interaction of smartphone user behavior and batteries [2]–[4]. The scale of the studies ranges from a couple of hundred [2] to twenty thousand [4] users or devices. However, existing empirical studies on large-scale users have mainly focused on devices and users. *To the best of our knowledge, there have been no existing large-scale user studies on application-battery interactions.* We believe one of the reasons for lacking of this kind of study is that it is difficult to calculate or predict how apps consume energy in the wild.

On the other hand, there exist many work on app energy modeling and optimizations for smartphones [5]–[8], but typically in a controlled environment with a limited number of users and apps. Although these smaller-scale modeling work consider app-specific energy issues, it cannot reveal the diversities of thousands of apps available in various app stores. In order to tackle this challenge, *our study focuses on understanding how different apps consume battery on smartphone devices based on large-scale battery traces*.

Compared to small-scale studies that can apply instrumentation and more complicated techniques while focusing on a small number of apps, large-scale studies are difficult to acquire fine-grained data. For example, a lightweight battery monitoring tool without modifying the Android operating system can only record battery changes at one percentage granularity. It is very difficult to calculate accurate app energy consumption rate using this kind of coarse-grained battery traces. We believe that this is one of the key reasons why earlier large-scale studies have been focused mainly only on user-battery interactions, instead of application-battery interactions.

This paper applies an approximate method to calculate app energy consumption rate based on coarse-grained battery traces. Although the battery traces of each app on a particular smartphone is inaccurate, we could achieve much more accurate results when we add thousands, or even millions, of traces together and calculate an average energy consumption rate for each app for all users. In this calculation method, we rely on the large amount of data available to amortize the errors in each specific trace. We evaluate this method with measured power numbers using the Monsoon Power Monitor [9]. Results show that the difference between the calculated energy consumption rate and the measured power current is less than 10%, which shows that the estimation method is pretty accurate with large-scale coarse-grained battery traces [10].

With this estimation method, we perform a large-scale battery study on smartphones focusing on diversities in apps and users. We have collected app and battery traces on over 120,000 Android smartphones for a four-week period. After filtering unusable data due to a variety of reasons, we analyzed the data on over 80,000 smartphones in this study. We analyze the types of apps, the battery consumption patterns for different types of apps, and distinctive characteristics for heavy and light users. Based on the analysis, we present our observations and discuss possible improvements on smartphone designs and mobile app development.

Among the many interesting findings in our study, the following are of particular interests.

- Only a very small percentage of smartphone time and battery is actually used for the traditional ''*cellphone*'' purpose. On average, 5% of battery and 6% of usage time are spent on voice calling. This not only confirms our assumption that smartphones are used for a variety of purposes, but also shows that smartphones are not ''phones'' anymore, instead should be called a mobile mini-computer with voice calling features.
- Power consumption during standby is actually much higher than expected. Our results show that the average power consumption during screen-off is 10X higher than the ideal idle state, which suggests that there are a lot of things going on in the background that causes standby time of many smartphones being reduced to less than a day.
- Despite the widely circulated complaints on the short battery life of many smartphones, we found that most phones could be used for more than one day. Considering the fact that many users are willing to (and able to) charge their phones at least once every day, the functionalities of smartphones have weighted more than the shorter battery life.

We make the following main contributions in this paper:

- To the best of our knowledge, this is the first large-scale study exploring the relationships between different types of apps and their battery consumption patterns. Although there have been many large-scale user studies on smartphone batteries, they mainly focused on smartphone devices and users because it is difficult to calculate energy consumption of each mobile app in the wild.
- We introduce a statistical method to calculate energy consumption rate for apps based on coarse-grained battery traces. We have shown that the calculation achieves accurate approximation compared to measured power numbers.
- Based on the result analysis, we present a list of observations, which we believe is helpful for identifying the special characteristics of smartphones from an application perspective.

The rest of this paper is organized as follows. We first introduce the data collection methodology in Section II and energy calculation method in Section III. In Section IV, we discuss an overall analysis on the users and apps studied. Then we perform a classification of apps in Section V and present the energy distribution and comparison of each app category. In Section VI, we perform a diversity study on different smartphone users. We discuss our observations and findings in Section VII. Finally, we present the related works in Section VIII and conclude the paper in Section IX.

II. DATA COLLECTION METHODS

In this section, we present an overview of the data collecting methodology used in our study.

A. CHALLENGES

In contrast to smaller-scale studies on app and battery in previous work, a large-scale study involving more than 100,000 users would face many challenges.

Although smaller-scale studies could manipulate the apps and ask users to cooperate, a large-scale study requires that the collection process is unobtrusive, such that users would not experience any service disrupt or performance degradation, with no significant network traffic increase. This requirement posts a strict limitation on the range and size of the collected data.

Since we collect information in the wild, we should be able to handle all the unusual problems occurring in the data collection process, such as user changing batteries or SIM cards, exceptions of data collection service or smartphone devices, irregularities in specific types of devices, etc.

B. DATA COLLECTION SERVICE

We developed a light-weight data collection service, which is integrated into an Android app called ''Energy Saver''

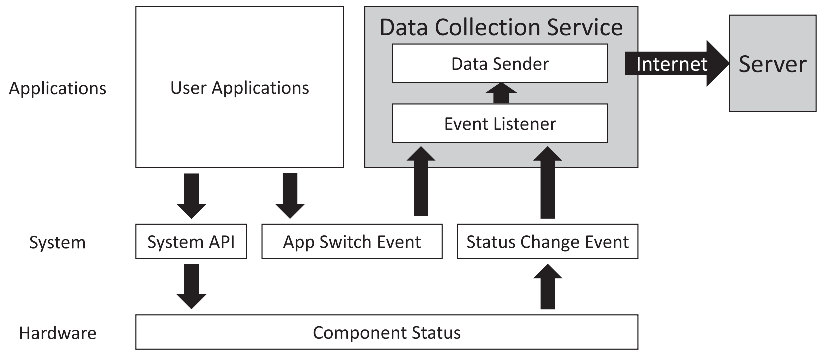


FIGURE 1. **Architecture of the data collection service.**

that once had more than a million active users.[1] Figure 1 presents the architecture of our data collection service, which contains two major components. The *event listener* collects all the events related to component status and apps, which is broadcasted in the Android system. The *data sender* is responsible of sending the event information to the server via the Internet. After the server receives the information, it will perform some processing work and store it into the database.

The service would run automatically after the smartphone is powered on, capturing changing status of battery levels, screen brightness, WiFi, Bluetooth and running app information. The collected data is first written into a local file. The local data will be transmitted back to the database on a central server periodically when the phone is connected to the Internet. A typical user would transmit less than 20KB data to the server in a single day, which would not introduce significant increase in their network traffic.

We collected data from over 120,000 Android users over a period of four weeks. The total data size is over 80GBytes. After incomplete and unusable data due to different reasons are filtered (which will be explained later), in the end, we use data from 80,000 smartphone users to perform the study.

1) COLLECTED EVENTS

We collect information broadcasted during events that cause interrupts or system calls, including battery level changes, app starts, screen on/off, network on/off, etc. At each event, we record the relevant data that represents the state changes. The information we collected include:

- *Battery level*: The battery level is a percentage of the whole battery capacity range from 0% to 100% that denote current remaining power of the smartphone.
- *Screen status*: Screen events have two types. One is the *screen on/off* event; the other is *brightness level* change, which is denoted by an integer ranging from 0 to 255.
- *Foreground app*: This information records the current app in use. We collect all app start events and foreground app switch events, such that we are aware which app is in the foreground at each time point.
- *Background apps and services*: Collecting this information can be very large in data size. We record them incrementally. When a smartphone is powered on, we start our data collecting service and record the list of background apps and services. After that, we only record the changes to the list of background apps and service.

2) DEVICE INFORMATION

In order to distinguish between different smartphone devices, we also need to identify smartphones and record device information such as the phone type, phone model, ROM version, etc. We identify smartphones by IMEI[2] and users by IMSI.[3] Both IMSIs and IMEIs are anonymized to protect the privacy of the users. Although most of the IMSIs and IMEIs have a one-to-one correspondence relation, we found that some IMEIs might correspond to several IMSIs. This kind of abnormal data will be filtered before we perform calculations.

Because the battery capacities might vary for different phone models, it will be inaccurate to represent the power dissipation rate based on battery level percentages collected from the system. When comparing the discharge rate between different phone types or different apps, we must take battery capacities into consideration. We collected the battery capacity data based on the standard configuration of each phone[4] and use this information together with the battery percentage drops.

C. DATA FILTERING

Since we collected the user data in the wild, there exists a small fraction of corrupted or incomplete data, which cannot

[1] The actual name of the app is "ShengDian Bao" in Chinese, which is a comprehensive energy saving app on Android. We collaborated with the company who developed this app to incorporate our collection service in an earlier version. As with many energy saving apps, users are asked and willing to share their battery traces to help improve the functions provided by the app. We have collected only privacy-unrelated data and these data have been used solely for research purposes.

[2] International Mobile Equipment Identity

[3] International Mobile Subscriber Identification Number

[4] Although a smaller fraction of users might use a different battery, we do not consider this as a common practice.

be used in our analysis. After the data are collected, we first filter out the invalid/unusable data according to the following heuristics.

1) OVERALL FILTERING

First, we filter out the invalid users using the following steps:

- *Unknown phone type*: We discard the users with invalid or unknown phone types, because we could not get the battery capacity info of these phone types to perform energy related calculations.
- *Unknown or uncertain users*: We discard the users whose IMEI numbers or network MAC addresses are invalid. We also remove the users whose IMSI corresponds to more than one IMEIs. Likewise, we remove the phones whose IMEI corresponds to more than one IMSIs. The reason is that the data from one of these IMEIs (or IMSIs, MACs) might be produced by several users and smartphones. If we put the event data from different users together, they could be confused by each other.
- *Minimum data collecting period*: We discard the users whose data collecting period is less than 2 weeks (The collecting service could be switched off or the hosting app could be uninstalled by the user).
- *Minimum number of events*: We remove the users whose events number is less than a pre-set threshold (set as 500 in this study).
- *Other abnormal users*: We remove the users whose devices record abnormal system time change, the users who frequently change batteries, etc.

2) APP-RELATED FILTERING

We define *an app usage period* as the time interval from the app starts or switched to the foreground, to the app ends or switched to the background. Within an app usage period, we store the time period of this usage event and the battery power change from the app start to the app end.

When we perform calculations related to power consumption, it is necessary to perform a pass to filter the inaccurate data, which consists of the following steps:

- *Minimum app using time*: If an app running in the foreground for less than 10 seconds, we will discard the data of this event.
- *Maximum idle time*: If a user record no events in more than ten hours, we regard the user's phone had powered off. We would reset and restart the calculation.

III. ENERGY CALCULATION

In this section, we present our method to perform energy calculation with the collected data, after filtered using the previously explained method.

A. CHALLENGES

While calculating energy consumption rate for each app, we face many challenges:

- One of the major obstacles is the granularity of the battery traces. Because we could only record battery changes at one percentage granularity, many short app traces consumes *zero* percent battery. In this case, many shorter app traces using less than 1% battery cannot be detected. We will show later that we could use a statistical method to calculate a combined average energy consumption rate for all users.
- When individual apps are concerned, multiprocessing becomes another major obstacle. As there are always multiple processes running at the same time, we are unable to distinguish the energy consumed by the foreground and background apps. As a simple solution, we only consider the foreground app as the current app consuming energy. In this case, because we are able to catch all process switching events, we always know which app is running in the foreground.[5]
- Apps might be used when the battery is charging. Thus we will distinguish between battery charging and discharging, and remove the app events during battery charging while calculating energy consumption rates.
- The data collecting service might affect the energy numbers because it also consumes energy. Because it is very difficult to estimate the energy consumed during data collection, we assume that the data collecting service does not consume any energy. We did not measure the energy cost of the collecting service because it was integrated into another Android app. However, because we only intercept the system intents that are broadcasted, no extra querying or modifying operations are actually conducted, thus the energy overhead is minimal.

B. CALCULATION METHOD

In order to calculate energy consumption statistics based on the recorded traces, we need to identify and filter out the events occurred during the charging periods. Then we can compute average energy consumption rate for each app based on battery trace collected from all user devices. We can also calculate projected battery life based on the energy consumption pattern of all apps.

1) BATTERY LIFE CALCULATION

We record the history of battery levels for each user with the battery level change events. We first separate the battery level history into charging regions and discharging regions to analyze charging habits of users and battery life of smartphones.

We consider a charging region as a battery charging event, including charging by A.C. electric sources and USB connections. If a fully-charged smartphone is still connected to an electric source, we will not consider this time period as a charging event. To analyze user charging behavior, we record

[5]When there exists an app running in the foreground, the background services and activities typically consumes much less power. However, when there are no foreground apps running, the background services still consume significant energy in a long period of time(We will demonstrate this later in our results.).

relevant information for each battery charging event, such as the time when a charging event starts/ends, the battery level when a charging event starts/ends, etc.

We calculate the battery life of smartphones based on the discharging regions of the battery level history. First, we can calculate the average discharging rate of each user using the following equation:

$$AverageDischargeRate = \frac{TotalBatteryDrops}{TotalDischargingTime}$$

Then we can calculate the projected battery life as follows:

$$BatteryLife = \frac{100\%}{AverageDischargeRate}$$

Note that we do not consider the battery capacity in this calculation. The discharge rate here is denoted by the battery percentage level.

2) ENERGY CONSUMPTION RATE FOR APPS

We calculate the power consumption of apps based on statistics with large-scale battery traces to reduce the error during calculation. First, we calculate the running time and battery level drops of every battery trace, then simply combine them together to calculate the *TotalUsingTime* and *TotalBatteryLevelDrops*.

As defined earlier, an *app usage period* is the time interval from the app starts or switched to the foreground, to the app ends or switched to the background (or the smartphone is powered off).

The average energy consumption rate of each app can be calculated as follows:

$$AvgEnergyConsumeRate = \frac{TotalBatteryConsumed}{TotalUsingTime}$$

Besides the coarse-grained battery level issue, we deal with the other challenges described above as follows:

- To deal with the charging influence, we detect the charging events using the battery level change events. We consider that the mobile device is charging if the battery level goes up. When calculating the power consumption, we only consider the running periods without charging.
- To deal with the background battery consumption, we regard all the battery level drops as consumed by the foreground app. It is acceptable because foreground apps usually use the CPU, screen and other components much more than the background apps and system services, therefore consuming more energy. We consider the background battery consumption as the general running environment of the foreground app in this study.
- Because of the timestamp problem, when we calculate the battery level drops, if we use the battery level of the nearest battery level change events to represent the battery level of the app switch event, there may be some errors for each battery trace. However, since the battery levels always change by 1%, the error of one battery trace is smaller than 1%. We expect that the error could be reduced to a tolerable level by the statistic method with large-scale analysis.

C. EVALUATION OF THE ENERGY CALCULATION METHOD

Although directly using each of the battery traces to calculate the power consumption would be inaccurate, we expect that the large-scale analysis could reduce the error into a tolerable level. Thus we perform a series of evaluation of different scale to evaluate our calculation method presented above.

Since the time duration of battery traces vary from several seconds to several hours, it is inappropriate to represent the scale of battery traces by the number of battery traces. In our evaluation work, we denote the scale of battery traces by the number of users.

In our evaluation, we select several different scales of users to calculate the power consumption, from 10 users to all the 80,000 users. We randomly select corresponding number of users for 5 times at every scales, from 10 users to about 10,000 users. Then we calculate the average and standard deviation of the results of each scale.

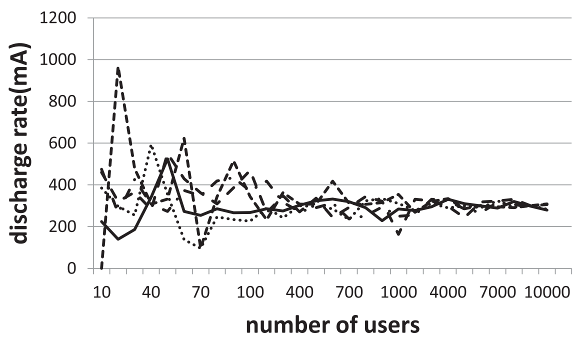


FIGURE 2. **Calculated power of *phone call* at different user scales.**

Figure 2 shows the results of calculated power consumption of phonecall with different scales of battery traces. It is obvious that the variance becomes smaller when the scale of battery traces increases. Then we analyzed the average, variance and the standard deviation (shown in Figure 3(a) and 3(b)). At the scale of 10 users, the calculated power consumption vary from 0 mA to 474.55 mA, while converge to a variance of 278.89 mA to 309.13 mA at the scale of 10,000 users. In comparison, we also calculated the power consumption with all the 80,000 users. The results of all users falls into the range of the variance of the result of 10,000 users.

We also conduct similar experiments for two other applications, Figure 3(c), Figure 3(d), Figure 3(e) and Figure 3(f) show the results of AngryBirds and SinaWeibo. We can also see that the energy consumption rate of each app converges to a fixed number when the number of users becomes larger and larger.

Overall, the results show that the power consumption rate of each app converges after the number of users reaches 10,000, thus we are able to achieve an accurate estimation with the combined average in our experiments. It indicates

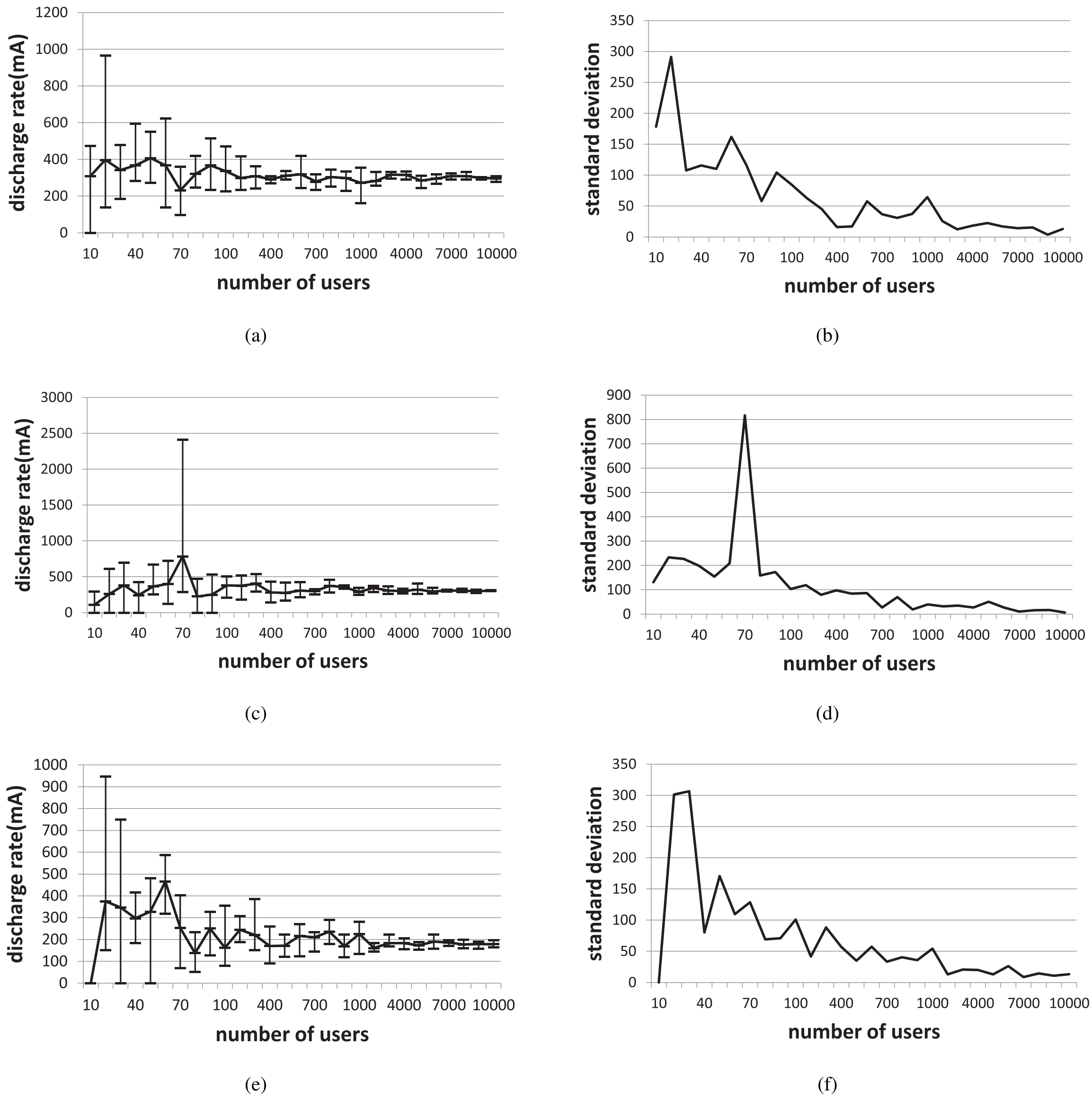


FIGURE 3. **Estimation results at different user scales for *phone call* (a)(b), *Angrybirds*(c)(d) and *Sinaweibo*(e)(f). (a)(c)(e) is the average power consumption and the variance at different user scales. (b)(d)(f) is the standard deviation at different user scales.**

that a relative high number (up to 10,000) of users is needed in order to provide accurate power estimation in this kind of study.

D. COMPARISON WITH MEASURED POWER NUMBERS

In order to further confirm the accuracy of our calculation method, we build an experiment environment using the Monsoon Power Monitor [9]. We perform measurements on a set of representative benchmarks from different categories using the power monitor. The measured data is compared to the calculated power numbers to check their accuracy.

The smartphone we measured is a Tianyu W806 device with dual-core Nvidia Tegra2 processor at 1GHz and 4.3 inch screen. The phone model is representative of the mainstream smartphones in the market during our data collection period.

Table 1 shows the comparison between average measured power current and average energy consumption rate calculated in our study. The difference between the measured data and the calculated power numbers are very small, with all differences less than 10%. Although this is only done for a particular device, it demonstrates that although our data collection and energy calculation method might not be accurate enough to calculate energy numbers for individual users with limited data, It is good enough to estimate the energy consumption rate of different apps over a large-scale data.

IV. OVERALL RESULTS AND ANALYSIS

We first present the battery statistics based on the data collected on 80,000 valid users and compare the results with earlier work.

TABLE 1. Comparison of measured current and calculated energy consumption rate of apps.

App	Measured current (mA)	Calculated energy consum. rate (mA)	Diff.
phonecall	276.83	295.96	6.91%
fruitninja	291.49	278.38	-4.50%
qiyi video	272.08	246.45	9.42%
estrongs	196.89	202.74	-2.88%
tencent WBlog	195.96	191.30	2.43%

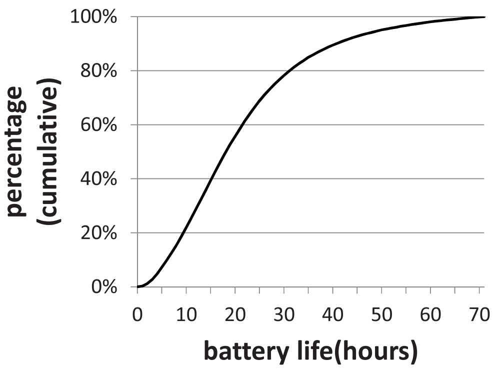


FIGURE 4. Distribution of battery life of all smartphones (CDF). The average battery life is about 22.6 hours.

A. BATTERY LIFE

The distribution of battery life of all users is shown in Figure 4. We notice that the battery life of about 75% smartphones can last more than 12 hours, with an average battery life of about 22.6 hours. It suggests that most smartphone users could keep their battery alive if they can charge once or twice every day.

How many times do batteries die? We attempt to identify a battery dead event if a low level of battery (less than 5% in this study) appears before a relatively long period of time (more than 1 hour in this study) in which the phone is not used. The result shows that batteries died for about 80,000 times for all 80,000 users within the four week period. On average, batteries died roughly once in a month per user. It shows that although batteries lasting less than a whole day for many users, they are able to charge the battery in time such that phone availability is rarely affected due to battery.

In Ferreira's study [11], all participantsąŕ devices are on for at least up to a full day. The likelihood of having a device on for up to two days is 33%, 18% for up to three and 11% for up to four days. However, their work counted all the power-off and reboot events, which could be caused by some other reasons. We only consider the long period of power-off events after a low battery level, which we think more likely to be battery died events.

B. BATTERY CHARGING HABITS

Battery charging is also an important activity of smartphone users. We study user charging habits from the following aspects.

- Figure 5(a) and Figure 5(b) show the distribution of battery levels at the start and the end of each charging period, respectively. It shows that the battery levels are evenly distributed when charging starts, while half of the battery levels of smartphones are more than 90% when charging ends. The result indicates that users consider little about their battery level when they decide to charge and rarely stop charging before their batteries are fully-charged to gain a longer battery life.
- Figure 5(c) presents the distribution of the charging periods. It shows that more than 80% of all charging events last for less than two hours. The results suggests that smartphone users tend to charge their batteries more frequently by a relatively short time period.
- Figure 5(d) presents the average charging frequencies for all users. It shows that 70% of the users are able to charge once or more per day on average. About half of the users charge their batteries more than twice per day on average. Considering this together with the battery life data earlier, we could conclude that most of the smartphone users can always get their batteries charged in time. Although smartphone batteries last less than a day, most users are able to charge their phones in time.

In comparison, Ferreira's study [11] pointed out that users mostly avoided lower battery levels (lower than 30% in their work). However, we could see in Figure 5(a) that the battery levels when charging events start are evenly distributed. There are also a significant number of users charging their phones at a low battery level.

V. DIVERSITY IN APPS

Because the apps running on smartphones are more and more complex, it is important to understand the behavior of different apps and how batteries are consumed by different apps.

A. APP CLASSIFICATION

We collected information on more than 23,000 different Android apps in our study. In order to understand the behavior of these apps better, we classify the top 200 apps based on total usage time into 7 categories (shown in Table 2). Each category represents a set of apps performing similar functions or using similar components of the smartphone, such that we expect them to behave similarly when user activity and energy dissipation are considered. Although these 200 apps only account for less than 1% of all the apps, the running time of these apps accounts for 82% of all the running time. Thus these 200 apps can be considered as a good representative.

Standby can be considered as a special "app" category, which represents the time period when users do not use their smartphones explicitly. We identify these time periods with the screen off status. However, standby in a smartphone is not the same as idle. When a smartphone is in standby mode, there might exist some apps and services or some network data transfer activity running in the background.

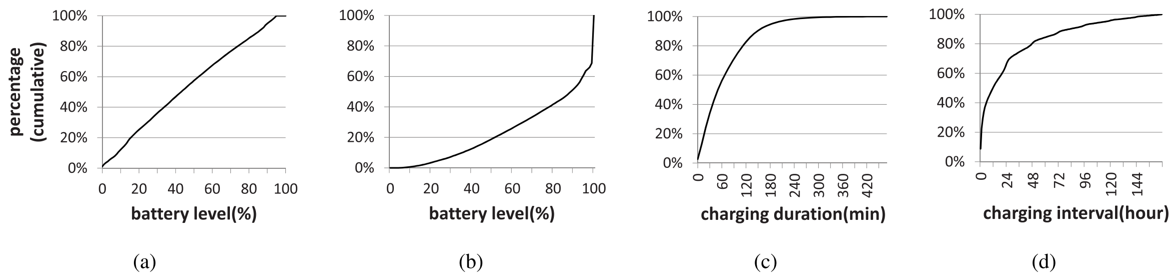


FIGURE 5. Charging Habits. (a) Battery level when a charging event starts. (b) Battery level when a charging event ends. (c) The duration of charging events. The average charging duration is 1.2 hours. (d) Distribution of charging intervals. On average, users charge their batteries every 27.3 hours.

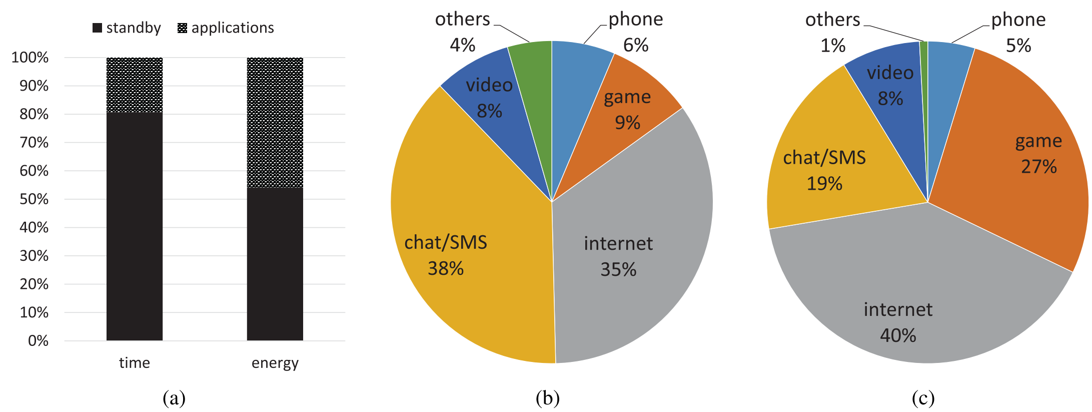


FIGURE 6. App Distribution. (a) Time and energy proportion of standby and apps. (b) Using time proportion of app categories. (c) Energy consumption of app categories.

TABLE 2. Categories of apps.

Category	Description	Examples
phone	voice phone calls	phonecall
game	smartphone games	angrybirds
internet	web app, navigation	ucbrower
chat/SMS	chat tools, SMS	minifetion
video/camera	video player, camera	kasend_video
others	others	estrongs
standby	idle, background	standby

Thus standby states behave differently in energy dissipation compared to real idle states.

B. APP DISTRIBUTION

First of all, we show the time and energy of a smartphone in use compared to when it is in standby in Figure 6(a). It shows that users use their phones for about 20% of the total time (around 4.8 hours each day), while consuming 45% of the total energy. More than half of the energy are consumed during standby while users are not explicitly operating on the phones. It indicates that even when the phone is in standby mode, there are still many activities going, such that about half of the battery are consumed during standby. We will analyze this phenomenon further in Section V-C.1.

Next, we consider the different categories of apps used. Figure 6(b) and Figure 6(c) present the distribution of each app category based on their total use of time and energy consumption. Among these app categories, the most used apps are chat tools and Internet apps, which occupy more than 70% in total usage time. Internet apps and games are more energy-hungry apps that consume 67% of the total energy consumption. We also notice that phone calls only record 6% of total usage time and 5% of total energy consumption, which shows that voice calling is far from the mostly used feature on a smartphone. A majority of people tend to use their smartphones as a "minicomputer" to access the Internet, play games and read e-books, instead of as a traditional feature mobile phone simply making phone calls.

In contrast, Bohmer's user study of about 4,100 participants [12] found that smartphones are still used mostly for text message and voice calling. But their work only considered the frequency of the usage. Our result shows that the usage time of text message and voice calling is much less than other smartphone apps.

C. ENERGY CONSUMPTION RATE OF APPS

To analyze the energy consumption patterns of different apps, we calculate an average energy consumption rate for each app category.

Figure 7 shows the energy consumption rate for each app category (including video, calling, sms, standby, etc.). The numbers here are adjusted to the default battery capacity for

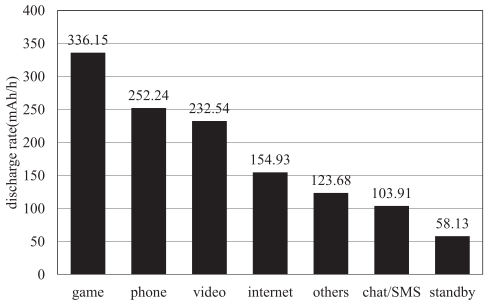


FIGURE 7. Energy consumption rate of app categories.

each phone model, thus all numbers are absolute consumption rate, instead of percentages obtained in the Android system.

We notice that apps related to gaming is the most power-consuming apps, with a 336.15 mA energy consumption rate. Other big power consumers are voice calling and video apps, whose energy consumption rates are about 252.24 mA and 232.54 mA, respectively. Chat tools and text messaging consume the least energy, the energy consumption rate is 103.91 mA. There is a 3-4x difference in energy consumption rate among these app categories.

1) STANDBY ENERGY

One surprising result is the power consumption rate for standby, which is more than 58mA on average. Based on our measurements, an ideal idle state with no activities normally consumes only around 5mA, which shows a 10X difference between an ideal idle state and the actual standby mode.

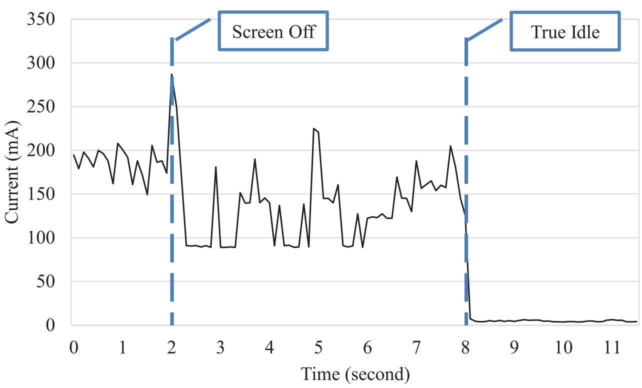


FIGURE 8. Measured power current after a user turned off the screen.

In order to understand the battery dissipation pattern during standby, we performed an experiment with the Power Monitor [9] to record the power current data, which is shown in Figure 8. We did not run any apps on the phone. The screen is turned off at the 2nd second in the figure. We can see that after the screen is turned off, the power did not go down immediately. It took about 6 seconds before the phone is actually "idle" and the current is stabilized at a low level.

Besides the activities right after screen off, other activities could still happen during standby, such as background services and apps, network data transportation, reading or writing the SD cards, etc. This suggests that the standby mode could still consume significant energy in smartphones, which could explain why the standby time of smartphones are much shorter compared to that of traditional feature phones.[6]

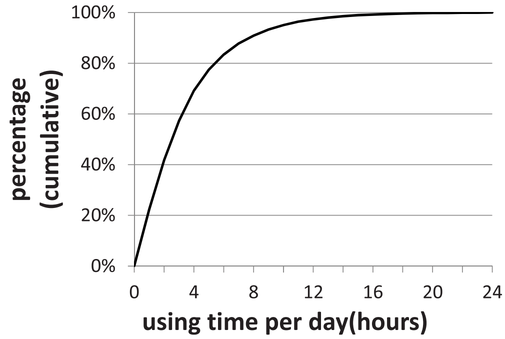


FIGURE 9. Distribution of smartphone using time per day for 50,000 users. The average using time per day for all users is about 4 hours.

VI. DIVERSITY OF MOBILE USERS

We have shown that mobile users use smartphones to perform a wide variety of tasks compared to traditional feature phones. In order to understand the behavior of different mobile users, we perform a diversity study to compare different users based on their usage time distribution.

A. USER-RELATED DATA FILTERING

Before conducting user diversity analysis, we need to identify user characteristics. In order to classify users based on their usage time, each user should have a relative long power-on time. Although we collect the traces from the previously evaluated 80,000 valid users during a four-week period, some of the users have a very power-on time that is not meaningful to perform further study. Thus we first perform some filtering work to remove these users. Specifically, we removed the users whose total power-on time are less than 5 days. The final number of remaining users in this diversity study is about 50,000.

B. USER CLASSIFICATION

To classify users based on their using frequencies, we calculate average using time per day for each user. This using time considers the total time of user apps such as video, games and phone call, etc. Figure 9 shows the distribution of average using time per day of the 50,000 users. We note that the variation of using time per day is from about 5 minutes for the least frequent users to about 20 hours for the most frequent users.

This result shows that the using time vary widely among different users. Based on this data, we classify the top 20% of the users as *heavy users*, while the bottom 20%of users are regarded as *light users*. Accordingly, the rest 60% of the

[6]With a 50mA average power and a 1500mAh battery, we can roughly calculate that the battery on a typical smartphone could last about 30 hours without heavy use, which is consistent to our observation.

users are classified as *normal users*. As a result, the heavy users use their smartphones more than 6 hours per day, while the light users use their smartphones less than 1 hour per day. On average, the using time of heavy users is 8.6 hours per day, while the using time of light users is about 0.9 hours. This gap in using time demonstrates the diversity of smartphone users.

In comparison to previous results, Falaki's study of 255 users [2] showed that different users interact with their phones 10-200 times a day on average; the mean interaction length of different users is 10-250 seconds. With some simple calculations, users use their phones 3.79 hours a day on average, which is very close to our result of 4 hours. Besides, Bohmer's user study [12] showed a result that mobile device users spend almost an hour a day using apps. This using time is just the level of light users in our study, while heavy users using apps for about 8.6 hours per day.

TABLE 3. App distribution of different user groups(*Heavy and Light users represent the top/bottom 20% of users based on their total using time per day, while the rest are categorized as Normal.*).

category	using time			energy		
	heavy	normal	light	heavy	normal	light
phonecall	1.67%	0.75%	0.82%	1.66%	1.93%	2.66%
game	2.03%	1.81%	0.66%	15.11%	13.39%	5.79%
internet	9.18%	7.29%	2.34%	24.34%	20.01%	7.38%
chat/SMS	10.91%	7.73%	2.46%	11.96%	9.45%	2.99%
video	2.69%	1.41%	0.45%	4.79%	3.76%	1.49%
other	1.91%	6.35%	0.18%	0.56%	2.54%	0.21%
standby	71.6%	74.7%	93.1%	41.8%	48.9%	79.5%

C. USER DIVERSITY IN APPS USED

We compare the distribution of using time and battery consumption among different app categories for three groups of users, respectively. Table 3 presents the comparison results.

For heavy users, the category "phonecall" ranks the last among the five app categories if we do not account for "standby" and "others". However, "phonecall" ranks in the third place for light users. This shows that heavy users are more intended to user their smartphones for various purposes other than voice calling.

Another interesting result is that while heavy users spent about 40% of their batteries on standby mode, light users spent almost 80% of their batteries on standby. This is because light users only use their phones for less than one hour on average, thus most of the battery are consumed during standby.

As expected, the using time and energy of normal users generally falls between the heavy and light users.

Soikkeli's user study of 140 users [13] showed that the smartphone usage is highly diversified across users, and thus an average user does not represent very well the people as a whole. Their work denoted the diversity of the differences of usage sessions among different users, while our work points out that the app distribution in using time among different types of users is also highly diversified.

D. USER DIVERSITY IN CHARGING HABITS AND BATTERY LIFE

because of the difference of app usage between heavy and light users, we want to check whether it will cause some differences in their charging habits and battery life.

Figure 10(a) presents the distribution of charging frequencies of heavy users and light users. On average, heavy users charge their batteries every 23 hours, while light users charge their phones every 29 hours. These results suggest that heavy users tend to charge more frequently. However, the results also shows that most users still need to charge their smartphones at least once every day, even when they only use their phones for less than an hour every day.

Figure 10(b) shows the battery life distribution of heavy users and light users. The difference of battery life between heavy users and light users is also obvious. The heavy users have a relatively higher discharging rate but a shorter battery life. On average, the battery life of heavy users is 21.2 hours, while the battery life of light users is 28.8 hours.

Although the difference in battery life is not so big compared to the average using time per day, it still shows a more than 30% difference. The explanation here is that most smartphones are running various background services, such that standby power is much larger compared to traditional feature phones, as shown in earlier results.

Oliver's study of 20,100 BlackBerry users [4] divided the users into 3 types by charging habits. The first type is the most aggressive energy consumers, consuming nearly 4.8% of their deviceąŕs energy per hour. Thus the battery lifetime is about 20.8 hours, which is close to our result of heavy users. They did not provide the exact average discharge rate of the other two types of user. However, the discharge rate of our light users is about 3.47% per hour, which seems to be a little higher compared to their other two types of user according to the figure in their paper.

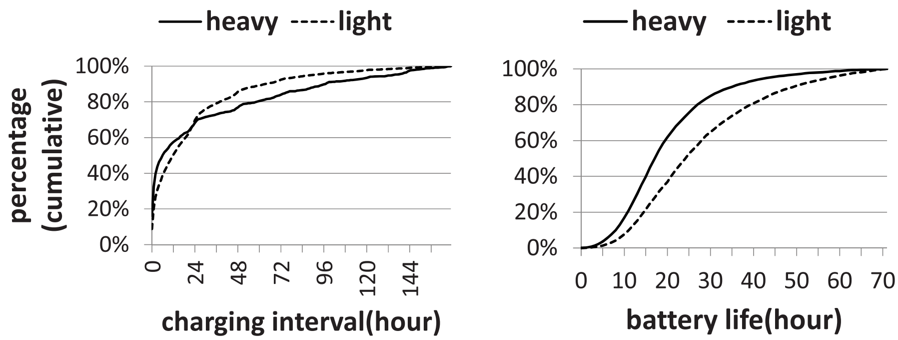
VII. DISCUSSIONS

Based on our analysis, we present some observations in this section and also discuss the limitations of our study.

A. OBSERVATIONS

We performed a large-scale study on smartphone apps and batteries on over 80,000 users in the wild. Although we could only collect coarse-grained battery traces with a lightweight service, we are able to calculate energy consumption rate for each app accurately with a statistical-based method. With the results and analysis presented above, we can draw the following important observations:

- *Only a very small percentage of smartphone time and battery is actually used for the traditional "phone" purpose.* On average, 5% of battery and 6% of usage time are spent on voice calling features. This confirms our assumption that smartphones are used for a variety of purposes, besides its traditional purpose as a phone.
- *Power consumption during standby is actually much higher than expected.* Our results show that the



(a) (b)

FIGURE 10. Heavy Users v.s. Light Users. (a) Charging frequency. On average, heavy users charge their battery every 23.3 hours, while the light users charge their battery every 29 hours. (b) Battery life. On average, the battery life of heavy users is 21.2 hours, while it is 28.8 hours for light users.

average power consumption during screen-off is 10X higher than the ideal idle state, which suggests that we should investigate what is actually going on when the phone is not used. We should also study optimization methods for not only active running apps, but also background activities. Although several recent work have attempted to optimize background activities and standby energy [14], [15], more efforts are needed to investigate it comprehensively.

- *We show that the charging habits of smartphones are different from feature phones, when users typically only charge their phones when battery capacity reaches a low threshold.* Due to the short battery life, smartphone users tend to charge their phones more frequently, but each charge lasting for a relatively short period.
- *Despite the widely circulated complaints on the short battery life of many smartphones, we found that most phones could be used for more than a day.* Considering the fact that most users accept the reality that many smartphones need to be charged every night, the functionalities of smartphones have weighted more than the shorter battery life.
- *Although there are thousands of apps in various markets, only a small fraction (i.e. 1%) of those apps are frequently used.* Researchers performing application studies (especially energy researchers) should turn focus to those apps used more frequently and consuming more energy.
- *The energy consumption rates of different apps differ greatly.* Energy researchers should try to understand the underlying differences between the most and least energy-consuming apps, and focus on analyzing and optimizing the more energy-hungry apps.
- *Smartphone users vary widely on using time of apps per day.* We chose a representative set of heavy users and light users to perform a user diversity study. We found that the two groups of users differ significantly in charging habits and the distribution of frequently used apps. Heavy users tend to charge more frequently, while having a shorter battery life.

B. LIMITATIONS

There exist some limitations in our study: some of them are inherent to this type of study, some are our choices in order to simplify the study.

For simplicity, we choose to attribute all energy consumed during a specific period to the foreground apps. Although we are able to retrieve the list of background processes and services, it is very difficult to distribute battery consumption to each of them. Neither did we consider the energy consumed by the operating system, because it is difficult to perform a more fine-grained analysis. Nonetheless, the energy consumption by background services and OS itself is very important and should be studied with further techniques.

For a large-scale study, we focus on app categories and how they consume battery power. We have taken into consideration some device information such as phone type and battery capacity. However, there are much more device features that could affect the battery consumption, such as screen sizes, standard CPU frequency, network access traffic, etc. We believe more detailed study on these factors could reveal many important results.

VIII. RELATED WORK

Energy consumption of smartphones and battery life issues have been extensively studied ever since smartphones have emerged. We present a brief overview of some closely related work in the following three categories: user studies on smartphone battery and app usages, studies on how apps consume energy, and smartphone energy modeling and optimization.

A. USER STUDIES ON SMARTPHONE BATTERY/APPS

Several large-scale user studies have focused on users and user-battery interactions. Falaki *et al.* [2] studied 255 smartphone users, characterized user activities and apps, and the impact of those activities on network and energy usage.

Banerjee *et al.* [3] designed a user- and statistics-driven energy management system, and conducted a user study, which shows that their system could harvest excess battery energy for a better user experience without a noticeable change in battery lifetime. Ferreira *et al.* [11] presented a 4-week study of more than 4,000 people to assess their smartphone charging habits to identify power intensive operations and to provide interventions to support better charging behavior. Oliver and Keshav [4] conducted one of the largest-scale study to measure the energy consumption of 20,100 BlackBerry smartphone users, and predicted energy level within 72% accuracy in advance.

There are also some work studying app diversity on smartphones. Shepard *et al.* [16] presented LiveLab, a methodology to measure real-world smartphone usage and wireless networks with a reprogrammable in-device logger designed for long-term user studies. They also present an iPhone 3GS based deployment of LiveLab with 25 users for one year to reveal different aspects of users and apps on smartphones. Böhmer *et al.* [12] performed a user study with 4,100 people for about 120 days to collect app usage data. They presented results on user behaviours including app usage over time, correlation of different app categories, and so on. Soikkeli *et al.* [13] detected end user contexts, and extracted smart phone usage session information from handset-based data of 140 smart phone users. Their usage session analysis found that smart phone usage is highly diversified across users.

These existing studies have revealed different aspects of the users and apps on the smartphones. However, no previous work have studied how batteries are consumed across different apps, which is the focus of our work.

B. STUDIES ON APP ENERGY CONSUMPTION

In order to study energy consumption of apps, many work have presented various energy models and conducted energy profiling and analysis based on these models.

1) ENERGY MODELING

CABLI [17] modeled the quantitative relation between system context attributes and the battery discharge rate based on multiple linear regressions. WattsOn [18] builds energy models based on hardware components of smartphones. AppScope [6] is an Android-based energy metering system to monitor hardware resource usage of an app when the app is running. The energy consumption of the app could be calculated through their hardware energy model DevScope [19].

2) ENERGY PROFILING AND ANALYSIS

Pathak *et al.* [5] performed a case study on the energy consumption of six popular smartphone apps with eprof [20], which could be used to perform detailed energy profiling using a system-call-based power model. They have found some issues in app energy consumption such as energy dissipation in third-part library, user data tracking and "wakelock energy bugs". In order to understand the energy consumption of background activities, EnTrack [21] is a system facility for analyzing energy consumption of Android system services.

3) ENERGY BUGS

A more comprehensive work on energy bugs [22] collected energy bugs from a few popular smartphone forums and bug reports, and classified the energy bugs into several groups. eDoctor [7] detected the energy bugs by finding the inconsistencies in resource usage and energy consumption when an app is running. eDoctor also recorded system events to analyze the reason of the abnormal energy consumption. Banerjee *et al.* designed an automated test generation framework that detects energy hotspots/bugs in Android applications [8].

Carat [23], [24] presents a collaborative method to diagnose abnormal energy drains of smartphone apps with about 400,000 users. They compared the average discharge rate among different apps and different users to detect energy bugs.

The experiments in most of these studies are typically performed in a controlled environment with a limited number of devices. Without large-scale user data, it will be difficult to gain real-world results on how batteries are consumed across different apps. Although Carat [24] is also able to measure app energy consumption rate in the wild, their main purpose is to diagnose energy issues and improve app energy consumption through a collaborative method, instead of analyzing the energy consumption issues for a large-scale users and mobile apps, which is the main purpose of this paper.

C. SMARTPHONE ENERGY OPTIMIZATION

Besides energy analysis and user study, many research work have proposed different techniques to reduce energy consumption of mobile systems from various angles.

One important direction is reducing CPU energy consumption. For example, E-MiLi [25] reduces the power consumption in idle listening with sleep scheduling. Catnap [26] reduces energy consumption of mobile devices by allowing them to sleep during data transfers. Song *et al.* [27] and Zhao *et al.* [28] use different strategies to scale down the CPU frequency in interactive applications.

For network energy optimization, TailEnder [29] reschedules network packets to reduce the "tail energy" and tradeoff with the network latency. Qian *et al.* [30] analyzed the periodic data transfers in apps with a trace contains 1.6 million network packets and compared several optimization technology such as fast dormancy, piggyback and batching. Perrucci *et al.* [31] optimized the cellular network energy by switch the network types between 2G and 3G. Xu *et al.* [14] optimized the email sync energy in standby mode by scaling the email size, inbox size and pull/push method. Ding *et al.* [32] modeled the relation between wireless signal strength and the network energy and optimized the network energy with a network-quality-aware method. ADEL [33] used taint analysis on app source code to find unused data downloaded from the internet.

For screen/display energy optimization, Lin *et al.* [34] proposed a video backlight scaling method via tradeoff between energy consumption and user experiences. FOCUS [35] saves display energy by dimming the less focused screen areas. Dong *et al.* [36] modeled the screen energy by pixel colors and optimized energy by changing theme colors.

In order to help users understand the power consumption, Chon *et al.* [37] proposed a system to pinpoint the major causes of battery drain in terms of both hardware and software aspects and phone configuration to extend application usage times.

Although this paper is not dealing with energy optimization directly, we hope to identify more optimization opportunities based on the analysis.

IX. CONCLUDING REMARKS

We have performed a large-scale battery study on over 80,000 smartphones and presented several sets of results related to apps and batteries based on our collected data. We have applied a method to calculate energy consumption rate of each app based on large-scale coarse-grained battery traces. Our data analysis found many interesting statistics and facts on app usage patterns, app energy consumption rate and mobile user diversity. We have also presented some observations based on the analysis results.

We believe that this type of study is able to reveal more information that could not be otherwise discovered in a small-scale lab study on limited number of devices. Our future work include performing further data analysis and exploring possible optimizations based on these findings.

REFERENCES

[1] (2013). *New Research Reveals Mobile Users Want Phones to Have a Longer Than Average Battery Life*. [Online]. Available: http://www.catphones.com/media/6638/Cat-Rugged-Research.pdf

[2] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 179–194. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814453

[3] N. Banerjee, A. Rahmati, M. Corner, S. Rollins, and L. Zhong, "Users and batteries: Interactions and adaptive energy management in mobile systems," in *Proc. 9th Int. Conf. Ubiquitous Comput. (UbiComp)*, 2007, pp. 217–234.

[4] E. A. Oliver and S. Keshav, "An empirical approach to smartphone energy level prediction," in *Proc. 13th Int. Conf. Ubiquitous Comput. (UbiComp)*, 2011, pp. 345–354. [Online]. Available: http://doi.acm.org/10.1145/2030112.2030159

[5] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? Fine grained energy accounting on smartphones with eprof," in *Proc. 7th ACM Eur. Conf. Comput. Syst. (EuroSys)*, 2012, pp. 29–42. [Online]. Available: http://doi.acm.org/10.1145/2168836.2168841

[6] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha, "Appscope: Application energy metering framework for Android smartphone using kernel activity monitoring," in *Proc. USENIX ATC*, 2012, pp. 1–14.

[7] X. Ma *et al.*, "eDoctor: Automatically diagnosing abnormal battery drain issues on smartphones," presented at the 10th USENIX Symp. Netw. Syst. Design Implement., 2013, pp. 57–70.

[8] A. Banerjee, L. K. Chong, S. Chattopadhyay, and A. Roychoudhury, "Detecting energy bugs and hotspots in mobile apps," in *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng. (FSE)*, New York, NY, USA, 2014, pp. 588–598. [Online]. Available: http://doi.acm.org/10.1145/2635868.2635871

[9] (2012). *Monsoon Power Monitor*. [Online]. Available: http://www.msoon.com/LabEquipment/PowerMonitor

[10] C. Wang, F. Yan, Y. Guo, and X. Chen, "Power estimation for mobile applications with profile-driven battery traces," in *Proc. 2013 IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Sep. 2013, pp. 120–125.

[11] D. Ferreira, A. K. Dey, and V. Kostakos, "Understanding human-smartphone concerns: A study of battery life," in *Proc. 9th Int. Conf. Pervasive Comput. (Pervasive)*, 2011, pp. 19–33. [Online]. Available: http://dl.acm.org/citation.cfm?id=2021975.2021978

[12] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer, "Falling asleep with angry birds, Facebook and kindle: A large scale study on mobile application usage," in *Proc. 13th Int. Conf. Hum. Comput. Interaction Mobile Devices Services (MobileHCI)*, New York, NY, USA, 2011, pp. 47–56. [Online]. Available: http://doi.acm.org/10.1145/2037373.2037383

[13] T. Soikkeli, J. Karikoski, and H. Hammainen, "Diversity and end user context in smartphone usage sessions," in *Proc. 5th Int. Conf. Next Generat. Mobile Appl., Services Technol. (NGMAST)*, Washington, DC, USA, 2011, pp. 7–12. [Online]. Available: http://dx.doi.org/10.1109/NGMAST.2011.12

[14] F. Xu *et al.*, "Optimizing background email sync on smartphones," in *Proc. 11th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, New York, NY, USA, 2013, pp. 55–68. [Online]. Available: http://doi.acm.org/10.1145/2462456.2464444

[15] X. Chen, A. Jindal, N. Ding, Y. C. Hu, M. Gupta, and R. Vannithamby, "Smartphone background activities in the wild: Origin, energy drain, and optimization," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, New York, NY, USA, 2015, pp. 40–52. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790107

[16] C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum, "Livelab: Measuring wireless networks and smartphone users in the field," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 3, pp. 15–20, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1925019.1925023

[17] X. Zhao, Y. Guo, Q. Feng, and X. Chen, "A system context-aware approach for battery lifetime prediction in smart phones," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2011, pp. 641–646. [Online]. Available: http://doi.acm.org/10.1145/1982185.1982327

[18] R. Mittal, A. Kansal, and R. Chandra, "Empowering developers to estimate app energy consumption," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 317–328.

[19] W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, "DevScope: A nonintrusive and online power analysis tool for smartphone hardware components," in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, 2012, pp. 353–362.

[20] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proc. 6th Conf. Comput. Syst. (EuroSys)*, 2011, pp. 153–168. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966460

[21] S. Lee, W. Jung, Y. Chon, and H. Cha, "EnTrack: A system facility for analyzing energy consumption of Android system services," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2015, pp. 191–202. [Online]. Available: http://doi.acm.org/10.1145/2750858.2807531

[22] A. Pathak, Y. C. Hu, and M. Zhang, "Bootstrapping energy debugging on smartphones: A first look at energy bugs in mobile devices," in *Proc. 10th ACM Workshop Hot Topics Netw.*, 2011, p. 5.

[23] A. J. Oliner, A. Iyer, E. Lagerspetz, S. Tarkoma, and I. Stoica, "Collaborative energy debugging for mobile devices," in *Proc. USENIX HotDep*, 2012, pp. 1–6.

[24] A. J. Oliner, A. P. Iyer, I. Stoica, E. Lagerspetz, and S. Tarkoma, "Carat: Collaborative energy diagnosis for mobile devices," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, 2013, p. 10.

[25] X. Zhang and K. G. Shin, "E-mili: Energy-minimizing idle listening in wireless networks," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2011, pp. 205–216. [Online]. Available: http://doi.acm.org/10.1145/2030613.2030637

[26] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 107–122. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814446

[27] W. Song, N. Sung, B.-G. Chun, and J. Kim, "Reducing energy consumption of smartphones using user-perceived response time analysis," in *Proc. 15th IEEE Workshop Mobile Comput. Syst. Appl. (HotMobile)*, 2014, p. 20.

[28] X. Zhao, Y. Guo, and X. Chen, "Transaction-based adaptive dynamic voltage scaling for interactive applications," in *Proc. 14th ACM/IEEE Int. Symp. Low Power Electron. Design*, 2009, pp. 255–260.

[29] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf.*, 2009, pp. 280–293.
[30] F. Qian *et al.*, "Periodic transfers in mobile applications: Network-wide origin, impact, and optimization," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 51–60.
[31] G. P. Perrucci, F. H. Fitzek, G. Sasso, W. Kellerer, and J. Widmer, "On the impact of 2G and 3G network usage for mobile phones' battery life," in *Proc. Eur. Wireless Conf. (EW)*, May 2009, pp. 255–259.
[32] N. Ding, D. Wagner, X. Chen, Y. C. Hu, and A. Rice, "Characterizing and modeling the impact of wireless signal strength on smartphone battery drain," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Modeling Comput. Syst.*, 2013, pp. 29–40.
[33] L. Zhang, M. S. Gordon, R. P. Dick, Z. M. Mao, P. Dinda, and L. Yang, "Adel: An automatic detector of energy leaks for smartphone applications," in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, 2012, pp. 363–372.
[34] C.-H. Lin, P.-C. Hsiu, and C.-K. Hsieh, "Dynamic backlight scaling optimization: A cloud-based energy-saving service for mobile streaming applications," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 335–348, Feb. 2014.
[35] K. W. Tan, T. Okoshi, A. Misra, and R. K. Balan, "Focus: A usable & effective approach to oled display power management," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 573–582.
[36] M. Dong, Y.-S. K. Choi, and L. Zhong, "Power-saving color transformation of mobile graphical user interfaces on oled-based displays," in *Proc. 14th ACM/IEEE Int. Symp. Low Power Electron. Design*, 2009, pp. 339–342.
[37] Y. Chon, G. Lee, R. Ha, and H. Cha, "Crowdsensing-based smartphone use guide for battery life extension," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2016, pp. 958–969. [Online]. Available: http://doi.acm.org/10.1145/2971648.2971728

YAO GUO received the Ph.D. degree in computer engineering from the University of Massachusetts, Amherst. He is currently an Associate Professor with the School of Electronics Engineering and Computer Science, Peking University. His research interests include operating systems, mobile computing, and low-power design.

CHENGKE WANG received the B.S. degree in computer science from Peking University, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science. His research interests include mobile energy modeling and optimization.

XIANGQUN CHEN received the B.S. and M.S. degrees in computer science from Peking University. She is currently a Professor with the School of Electronics Engineering and Computer Science, Peking University. Her research interests include operating systems and software engineering.