One Charge for One Week: Hype or Reality?

Youhuizi Li, Bing Luo, Hui Chen and Weisong Shi
Department of Computer Science
Wayne State University
Detroit, USA
{huizi, luobing, huichen, weisong}@wayne.edu

***Abstract*—For mobile devices, battery energy is the most precious resource. In the last decade, researchers have proposed various energy saving strategies from the system level to the hardware component level. In this paper, we explore how close to one week a smart phone running off of a single battery can last under normal usage. We first developed a battery lifetime prediction model that considers the influence of both user behavior and hardware components. Through experiments we analyzed the assumptions and the accuracy of the prediction model. We discussed the error rate of the estimated applications' power as well as its influence on the battery lifetime prediction. To analyze the impact generated by user behavior, we classify users into six types based on their application usage pattern. The theoretical battery life and potential extended battery time for each user type, with and without hardware improvement, have been illustrated. For example, compared with the original 66h (2.75 days) for users who rarely use their smartphone, we found that the battery life can be extended to 147h (more than 6 days) when we only maintain applications in the top three commonly used categories. Finally, several aspects, such as sleep frequency and background applications, that may affect prediction results are discussed.**

Keywords*—*Mobile device; Battery life; User behavior

I. Introduction

Mobile devices are becoming more and more intrinsic in our daily lives. Global smartphone users will reach 1.75 billion in 2014, which is roughly one quarter of the world's population, and more than half of the population in the United States have their own mobile devices now [1]. However, the battery drain issue seriously influences the user experience and a survey shows that battery life is the single main gripe of today's mobile phone user [2]. To solve this problem, researchers have been trying to find optimization approaches to extend battery life. They have proposed optimization approaches from the operating system level, which require major modifications to be made to today's devices [3], [4]. There are other works that focus on specific components to improve energy efficiency [5], [6]. However, it is hard to evaluate the energy saving performance of these approaches. How to compare the influence of memory saving with GPS saving? How long can the battery lifetime be extended for each application after applying the approach? We are still missing a standard mechanism to evaluate the effectiveness of the saving approaches from the perspective of real device usage.

Aside from saving energy from the device itself, user behavior also affects battery discharging time seriously. Falaki *et al.* [7] collected two user groups' information to analyze the usage of the smartphones. They calculated the mean and standard deviation of energy that users drain in an hour. The results show that the battery usage is very different from user to user. The heaviest users drain close to 250 mAh, while the lightest of users drain only 10 mAh. Hence, users can extend the battery lifetime by adjusting their behavior, although they may not like to. Moreover, the performance of energy saving approaches mentioned above are different for each user. As a result, the user behavior needs to be considered as an important factor when we improve battery life of mobile devices.

In this paper, we undertake the following question: how close can one battery charge survive seven days for normal smart phone users. We first developed a prediction model that calculates how long the battery can be extended under various situations. The model takes hardware components' information and user behavior into consideration, providing a mechanism to evaluate various energy saving methods for different users. Assuming the application's power is relatively stable and the user behavior pattern is known, we can analyze the influence of each hardware component to the device's battery life. For some users, if the energy efficiency of the display is doubled, the battery lifetime will increase 18.57%. From the perspective of users, the possible maximum battery lifetime can be calculated as well. For example, compared with the original 66h for users who rarely use their smartphone, we found that the battery life can be extended to 147h (more than 6 days) when we only maintain applications in the top three commonly used categories. Moreover, given a target battery discharging time, the prediction model will provide the information for how much improvement we need to achieve. At last, several aspects that may affect prediction results are discussed.

In this paper, we have four main contributions:

- We propose a prediction model that estimates how long the battery life can be extended based on the user behavior and hardware usage.
- We verify the assumptions proposed in the model and analyze the accuracy of the prediction result in both theoretical and experimental ways.
- We study the user application usage data and classify users into several types. For each type, the theoretical maximum battery time is calculated.
- We discuss the influence of hardware improvement on the device's battery life and analyze the prediction results under several optimization cases.

This work is in part supported by NSF grant CNS-1205338, the Introduction of Innovative R&D team program of Guangdong Province (NO.201001D0104726115), and Wayne State University Office of Vice President for Research. This material is based upon work supporting while serving at the National Science Foundation.
978-1-4799-6177-1/14/$31.00

The remainder of the paper is organized as follows: We describe the prediction model in Section II. Then applications' power, user traces and model accuracy are analyzed in Section III. Section IV applies the model to each user category and predicts the battery life based on hardware improvement. Following that, we discuss several aspects that may affect model accuracy in Section V and present the related work in Section VI. Finally, Section VII summarizes the paper.

II. Battery LifeTime Prediction Model

In this section, we listed the assumptions that our prediction model is based on. Following that, we illustrated the power models that estimate average application power and the battery lifetime prediction model.

A. Assumptions

To simplify the problem, the applications and devices that we focus on should satisfy the following assumptions:
(1) The average power of each hardware component is relatively stable and is linear to the usage of the component.
(2) User behavior has specific patterns and the pattern can be expressed by the user-interacted applications' running time.

B. Lifetime Prediction Model

To calculate the battery lifetime, it is important to estimate the average power. Since user behavior is represented by the application running time, the power should be calculated at the application level. Table I presents the power models for the average power of the four main hardware components. They are abstract formats used to show that the power is linear to the average component's usage. For CPU power, the main indicator is the utilization which is calculated from the user time, kernel time and sampling interval. Since the power variation for different colors is not very large for the Google Nexus 4, we only consider brightness in display power. The radio and Wi-Fi power is based on the signal states and packets rate respectively. More details of the specific power models can be found in [8]. In the average power models, all c are constants and u are the usage of each component. We denote

$$\vec{M} = (m_{cpu}, m_{display}, m_{radio}, m_{wifi})$$

as the component power vector of one application, then the whole average power of the application is $a = sum(\vec{M})$. Now, suppose there are n applications, the average power of the ith application is a_i. Let

$$\vec{A} = (a_1, a_2, ..., a_n)$$

be the application power vector, and let

$$\vec{B} = (b_1, b_2, ..., b_n)$$

TABLE I. The power models for main hardware components.

Component	Average Power Model
CPU	$m_{cpu} = c_{cpu} * u_{cpu} + c_{cpuidle}$
Display	$m_{display} = c_{display} * u_{display} + c_{displayidle}$
Radio	$m_{radio} = c_{radioscan} * u_{rs} + c_{radioon} * u_{ro}$ $+ c_{radioidle} * u_{ri} + c_{radioactive}$
Wi-Fi	$m_{wifi} = c_{wifilow} * u_{low} + c_{wifihigh} * u_{high} + c_{wifiidle}$

be the percentage of time that application is used by a user, then the average power for this user is

$$p = \vec{A} * \vec{B}^T.$$

Further, suppose the battery energy is E, then the life time is:

$$t = \frac{E}{\vec{A} * \vec{B}^T} \tag{1}$$

If the energy efficiency of the ith component can increase x_i times, the battery energy becomes E'. Let $s_i = \frac{1}{x_i}$, and let

$$\vec{M'} = (s_1 * m_{cpu}, s_2 * m_{display}, s_3 * m_{radio}, s_4 * m_{wifi})$$

be the new component power vector and let $\vec{A'}$ be the new application power vector, then the whole life time can increase

$$r = \frac{t_{improved}}{t_{original}} = \frac{E' * \vec{A} * \vec{B}^T}{E * \vec{A'} * \vec{B}^T} \tag{2}$$

III. Model Analysis

In this section, we analyzed the assumptions of the prediction model and presented the relationship of application power estimation error rate and lifetime prediction model accuracy.

A. Experiment Setup

The mobile device used in our power related experiments is a Google Nexus 4, and its specification is listed in Table II. We use a BK Precision programmable power supply [9] to power up the smartphone, which provides a constant voltage of $3.8V$ and a maximum current of $3A$. It samples the current four times per second, and sends the data to the computer. To reduce the interference, we installed the target application on a clear OS and stopped all unnecessary services, such as Google Plus and Google Play services. Table III lists the applications used in the rest of the paper. Some of them are among the top 10 most popular applications on the Android Market.

For user behavior monitoring, we modified the *ActivityStack* class in the Android OS to log the *Pause* and *Resume* state with the corresponding time to get the application usage information. In addition, the battery capacity was recorded each time the screen was off. It presented the total energy consumption for the interactive session. We gave two Google Nexus 4 phones to 14 students who were randomly picked throughout the university. The device was their primary smartphone during the one week experiment time, they were free to install applications and modify the system configurations.

TABLE II. The specification of Google Nexus 4.

Component	Specification
OS	Android 4.3; kernel version 3.4.0
Chipset	Qualcomm Snapdragon APQ8064 S4 Pro
Processor	Quad-core Krait; 384 - 1512 MHz; L0: 4 KB + 4 KB, L1: 16 KB + 16 KB, L2: 2 MB;
RAM	2G ; Dual-channel 533 MHz LPDDR2
Display	4.7 in diagonal IPS; 1280768 px; 320 dpi
GPU	Adreno 320
Radio	Integrated 3G/4G World/multimode
Wi-Fi	Integrated digital core 802.11n (2.4/5GHz)

TABLE III. APPLICATIONS USED IN THE ANALYSIS.

Category	Application	Description
Photography	Gallery	View and edit photos.
Business	Amazon	Go over the popular items, choose several items to see the detailed info.
	Dealmoon	
News	BBC	Go over the popular news.
	SinaNews	
Travel	Yelp	Search nearby restaurants, see the photos.
Games	CandyCrush	Complete the first two levels.
	Temple Run	
	AngryBirds	
Phone	Phone Call	Make and answer phone calls.
Media	YouTube	Search a MV and play several mins.
	Youku	
Music	Pandora	Randomly choose a channel, listen for several mins.
	Douban	
Email	Gmail	Read 10 latest emails, send one email out.
	Default Email	
Weather	WeatherChannel	Search two cities' weather, check detailed info.
	YahooWeather	
Social Networking	QQ	Communicate through instant massage.
	Facebook	Go over several new states, then post news.
	Twitter	
Navigation	Map	Search road info from local to mall.
Utility	Calculator	Multiply and divide random numbers.

B. Application Power Stability

To build an accurate battery time prediction model, one of the assumptions we make is that the application power is relatively stable. We classify users into different types according to their application usage which leads to various user power behaviors. Generally speaking, to finish the same task, for example, editing a photo, the application power should be the same. We run several applications to perform the repeated tasks and monitor the device power variation. Fig. 1 presents the device power variation when we played with the *Gallery* application. After viewing two photos, we modified them by adding filters, cropping it, rotating it and saving it. As the figure shows, there are power peaks at the beginning of viewing. They are usually caused by user interaction (touch, click, etc.) and display rending. The power trend in the two viewing and the two editing are similar in manners and the differences in average power are 18 mW and 32 mW respectively. Compared with the 900 mW viewing power and 1850 mW editing power, the *Gallery* tasks' power are relatively stable. As the results show, the power behavior of performing identical tasks is relatively stable. For the same user, the applications' power are stable since the usage scenarios are the same. For instance, the user prefers viewing friends' information rather than posting status when using *Twitter*. The user evolution problem, for example, some users may like viewing photos for the first several months while later they may prefer to edit their photos, is out of the scope.

C. Application Power Accuracy Analysis

After presenting the application power is relatively stable, the next step is how to conveniently get the power. Similar to previous work [10], we leverage power models to calculate applications' power. From the prediction model, we can see that the accuracy of application power directly influences the prediction accuracy. We evaluated the power models by comparing the estimated power with the measured power. The experiments are done on the popular applications listed in Table III. The power error is defined as (3), which is different for each category. For example, the error rate of *Pandora* is

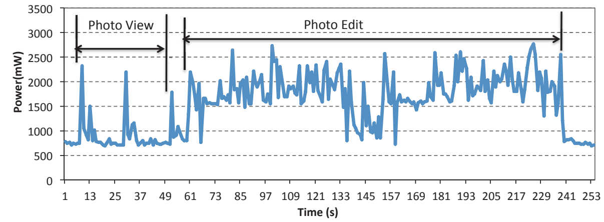

Fig. 1. The device power variation of Gallery.

11.73%, while it is 4.72% for *Facebook*.

$$error = \frac{|estimated - measured|}{measured} \tag{3}$$

We took the average error of all application categories as the power models' error. Fig. 2 shows the error information for the parts of the popular applications. The red part is the estimated power subtracting the measured power, so it may be negative for some applications. For *Media* and *Games*, the estimated power is usually greater than measured power and the error is around 10%. For *Social* and *Business* applications, for example, *Amazon*, the error is much smaller and it is about 2%. As a result, the average application power error of the power models is 7.31%.

D. Error of the Lifetime Prediction Model

Suppose the error of a_i (average application power) is e, which means

$$\forall i = 1, 2, ..., n. \quad |a_{iReal} - a_{iEstimated}| < e * |a_{iReal}|$$

Then, because

$$\sum_{i=1}^{n} b_i = 1$$

, and

$$\forall i = 1, 2, ..., n. \quad 0 \leq b_i \leq 1$$

So

$$|\vec{A_{real}} * \vec{B}^T - \vec{A_{estimated}} * \vec{B}^T| < e * |\vec{A_{real}} * \vec{B}^T|$$

Then the error of estimated life time is:

$$|\frac{t_{estimated} - t_{real}}{t_{real}}| = \frac{\frac{E}{\vec{A_{estimated}} * \vec{B}^T} - \frac{E}{\vec{A_{real}} * \vec{B}^T}}{\frac{E}{\vec{A_{real}} * \vec{B}^T}}$$

$$= \frac{|\vec{A_{real}} * \vec{B}^T - \vec{A_{estimated}} * \vec{B}^T|}{\vec{A_{estimated}} * \vec{B}^T} < \frac{e * \vec{A_{real}} * \vec{B}^T}{\vec{A_{estimated}} * \vec{B}^T}$$

$$< \frac{e}{1-e}$$

Since we can measure the current lifetime, and

$$r = \frac{t_{improved}}{t_{original}}$$

so the error of r is also $\frac{e}{1-e}$.

Figure 3 illustrates the trend of the battery life prediction error. When the application power estimation error is within 10%, the battery life prediction error is almost the same as the estimation error. When it is greater than 20% (battery prediciton error 25%), the prediction error is much larger than power estimation error. In our case, the error of battery prediction is less than 10%.

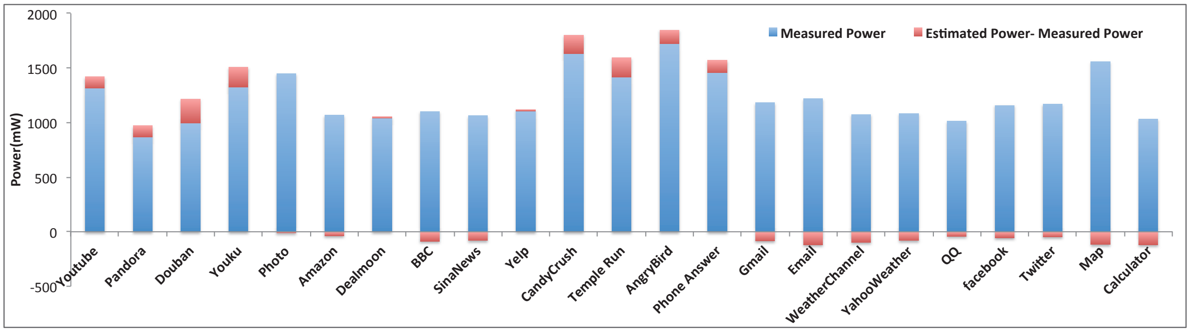

Fig. 2. The distribution of estimated power error for popular applications.

TABLE IV. AN OVERVIEW OF THE TWO DATASETS.

	#Users	Platform	Duration	#Apps	#Categories
Dataset1	34	iPhone 3GS	12 weeks	2400	26
Dataset2	14	Google Nexus 4	1 week	135	21

E. User Behavior Analysis

User behavior is one of the major factors that affect battery lifetime. There are two datasets used in the paper to analyze user behavior. Dataset1 is the subset of the LiveLab trace [11], which contains 34 iPhone 3GS users' usage information from October 2010 to January 2011. Dataset2 was collected by ourselves, it has 14 students' usage information for one week. Table IV summarizes the two datasets.

Dataset1 was used as training data and we classified Dataset2 users into different types according to their application usage. *Phone call*, *SMS* and *Sleep* were treated as normal applications for consistency. Since there are 2400 applications, we choose application categories as the cluster metrics. To identify natural groupings of the user behavior data, we applied Fuzzy C-Means (FCM) clustering algorithm [12], which gives the best results for overlapped datasets and comparatively better than k-means algorithm. The centroid of a cluster is the mean of all points in the dataset weighted by their degree of belonging to the cluster. The weight is updated every iteration as (4) shows, and the distance we used was Euclidean distance (5). The level of cluster fuzziness m was set to 2. All the data are normalized before clustering.

$$weight_k(x) = \frac{1}{\sum_{j=1}^{ClusterSize} \left(\frac{d(center_k, x)}{d(center_j, x)} \right)^{\frac{2}{m-1}}} \tag{4}$$

$$distance(k,x) = \left(\sum_{i=1}^{FeatureSize} (k_i - x_i)^2 \right)^{\frac{1}{2}} \tag{5}$$

The classification becomes more accurate as the size of cluster increase. We chose the size 6 because the improvement after 6 clusters is not as much as before. The corresponding application usage information for each cluster (user type) is demonstrated in Table V. We listed major application categories in the table, each data is the average percentage of the running time of all applications in the category over the total logging time per day. Most times the devices were in the sleep state, except for user type T_2 which usage was dominated by phone calls. T_2 only has 1.7% items, its data is generated mainly by one user who made phone calls a lot and barely played with other applications. Asides from *Sleep*, users spent more time on social networks, browsers, games, media and photograph applications. After applying the cluster information to dataset 2, the result show it covers four user types and each user belongs to two types on average during the one week. To further explore user behavior information, Fig. 4 illustrates the user type variation of one randomly picked user during 12 weeks. The user behavior in application usage has patterns that can be predicted but it is not simply fixed. Most of the time the user belonged to T_3, but he also changed to T_1, T_2, T_5 and T_6 occasionally. We can use recent history to predict or take a weighted average value as usage information. Figuring out an accurate user behavior prediction mechanism is not the focus of the paper. For simplicity, we assume user behavior information is known.

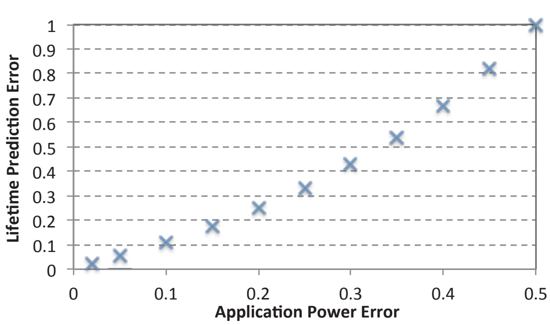

Fig. 3. The relationship of battery prediction error and application power estimation error.

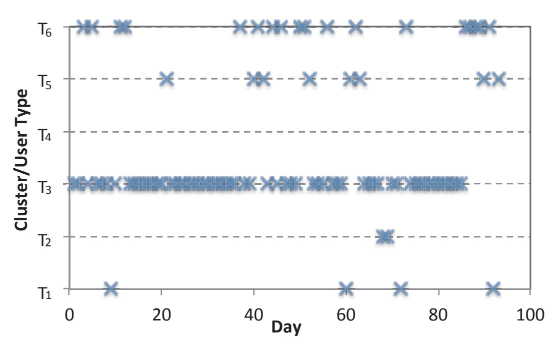

Fig. 4. User cluster type changes with time.

TABLE V. THE APPLICATION USAGE INFORMATION FOR EACH USER TYPE.

User Type	Utilities	News and Magzines	Email	Games	Media	Photography	Browser	Social Networking	Weather	Phone Call	Sleep	SMS
T_1	1.08%	0.49%	1.18%	1.75%	3.89%	2.56%	4.08%	4.69%	0.58%	5.27%	59.01%	6.47%
T_2	0.12%	0.05%	0.08%	0.08%	0.25%	0.16%	0.21%	0.21%	0.09%	96.27%	0.60%	0.82%
T_3	0.34%	0.06%	0.34%	0.62%	0.60%	0.73%	0.89%	1.12%	0.30%	1.21%	91.70%	1.36%
T_4	2.29%	0.94%	1.49%	1.69%	4.39%	2.08%	4.83%	6.57%	0.78%	7.89%	26.85%	8.22%
T_5	0.93%	0.32%	0.89%	1.71%	3.09%	2.23%	3.65%	3.53%	0.55%	4.12%	68.82%	5.07%
T_6	0.68%	0.20%	0.67%	1.30%	1.74%	1.75%	2.32%	2.41%	0.45%	2.71%	79.21%	3.52%

TABLE VI. THE DETAILED POWER AND TIME INFORMATION FOR APPLICATIONS TESTED IN THE EXPERIMENT.

Application	Estimated Average Power(mW)	History Average Power (mW)	Duration (mins)	Percentage
Temple Run 2	1541.69	1593.86	14	11.87%
Dealmoon	1281.06	1053.33	26	22.03%
YouTube	1233.44	1419.68	48	40.68%
Weather	924.88	974.47	2	1.69%
Twitter	1147.97	1125.48	4	3.39%
Photo	867.00	1440.71	9	7.63%
BBC News	875.65	1012.94	15	12.71%

F. *Prediction Model Verification*

To verify the prediction model can accurately calculate the battery lifetime, we simulated the normal usage cases and compared the estimated battery time with the real discharging time. In the experiments, all the parameters were logged twice per second. The running time of each application was predefined, and the detailed experiment information is illustrated in Table VI. As we mentioned above, the application power is relatively stable while different tasks consume different power. We took the average power as the application power, for example, we did not distinguish video playing power and video searching power for *YouTube*. According to the logged information, the average power of applications and their running time were calculated. Multiplying the applications' power and their running time percentage, the result 1206.1 mW was the average power in this experiment. The battery capacity was also recorded with the same frequency, it decreased from 72% to 40% and the full capacity is 2097 mAh. The battery voltage is 3.8V, so the energy consumed was 2549.95 mWh. According to the prediction model, this amount of energy should support 126.85 minutes. Compared with 118 minutes, the error of the prediction model is 7.5%. In reality, we cannot log the parameters in the background all the time since the monitoring program itself is energy consuming. At least, there is no such low overhead service that provides application real time power information. So, another option is leveraging history average applications' power rather than calculating power from real time parameters. The result shows the error was less than 1% in the same experiment.

The prediction model error is less than we calculated since the battery capacity is not very accurate. The voltage decreases as the battery capacity decreases [13], and in our calculation the voltage is 3.8V all the time. So the real battery capacity is less than 2549.95 mWh. Besides, the granularity of battery capacity is 1%. There was at most 209 mAh that we did not know if it was used or still reserved, which depends on the battery capacity update policy. Moreover, part of the error is caused by the power models that are used to estimate application power as Section III-D demonstrated. Hence, the prediction model is reasonably accurate to calculate the potential battery lifetime.

IV. APPLICATIONS OF PREDICTION MODEL

After validating the battery life prediction model, we analyzed the potential battery extended time for different user types according to the model and the future trend with the improvement of hardware components.

A. *The Theoretical Battery Lifetime*

For the same device, battery lifetime for different users are also different. We presented the theoretical battery lifetime for the six user types based on the typical application usage information and calculated the corresponding potential battery extended ratio by changing the user behavior.

Although the power of each application is different whether they are in the same category or not, it is impossible to gather every application's power and analyze the battery time for each combination. For our experiment, we classified users based on the time they spend on each application category. We used the average power of applications we tested to represent "category power", such as 1591 mW for *Games*, 1081 mW for *News*. In the specification for Google Nexus 4 [14], the battery capacity is 2100 mAh, the standby time, talk time, video playback and web browsing time are 250 h, 10 h, 8 h and 7 h respectively. Hence, we can calculate the corresponding power. For all other applications which do not belong to categories listed in Table V, we use 1000 mAh to represent their power. The summary of each category power is presented in the first row of Table VII. Hence, we have the value of $\vec{A}$ and $\vec{B}$ illustrated the in prediction model in Section II. The second row of Table VII is battery discharging time if the system only has the corresponding applications running. It gives the intuitive impression of the category power. The time varies from 5 hours to 10 hours with exception of *Sleep*.

Since making phone calls and sending messages are the basic functions of a phone, we did not modify their usage in the following analysis. For the T_1 type of users, the estimated battery time was 17.56h. The users spent nearly the same time on *Browser* and *Social Networking* applications, following that was the *Media* applications which occupy 3.89% of total time. If we keep these top three categories as well as *Phone Call* and *SMS* and count other applications' time on *Sleep*, the battery can last 29.6h. For users in type T_2, they treated the device as a basic phone and spent over 95% of the time on *Phone Call*. So there is not much time that can be extended. Similarly, user type T_3 also contains a category, *Sleep*, that is responsible for over 90% of the time. The battery time for the typical T_3 type of users is 66.28h since most of the time the device is in sleep state. If the user does not play any applications except *Phone Call* and *SMS*, the battery time will be extended to 147.3h. It is because the *Sleep* power is much lower than other applications and the *Sleep* power also dominated the energy

TABLE VII. THE CATEGORY POWER AND THE SUMMARY OF POTENTIAL BATTERY EXTENDED FOR EACH USER TYPE.

	Utilities	**News and Magzines**	**Email**	**Games**	**Media**	**Photography**	**Browser**	**Social Net-working**	**Weather**	**Phone Call**	**Sleep**	**SMS**	**r**
Power(mW)	1032	1081	1201	1591	997	1407	1140	1163	1057	798	32	981	
Time (h)	7.73	7.38	6.65	5	8	5.67	7	6.86	7.55	10	249.37	8.13	
User Type T_1					✓		✓	✓		✓	✓	✓	1.68
User Type T_2										✓	✓	✓	1.03
User Type T_3										✓	✓	✓	2.22
User Type T_4					✓		✓	✓		✓	✓	✓	2.24
User Type T_5					✓		✓	✓		✓	✓	✓	1.61
User Type T_6					✓	✓	✓	✓		✓	✓	✓	1.39

consumption in the original case , the battery can last much longer by increasing 5.73% time for *Sleep*.

For user type T_4, the top categories were *Social Networking*, *Browser* and *Media*. The time spent on *Social Networking* is 7.89%, while the other two occupied almost the same amount of time which is around 4.5%. After maintaining these three categories, the estimated battery life time was 23.43h. Compared with the former 10.45h, it doubled the battery lifetime. Similarly, users in type T_5 also preferred the three categories, while the percent is around 3.42%. The potential battery life was 13h longer than the typical case. Lastly, for user type T_6, there is one more category that should be considered aside from the top three since *Photography* and *Media* have the same percentage. The battery life can be increased up to 40%.

The summary of the potential extended battery life for each user type is also presented in Table VII. The r is the increased battery time over the corresponding original time as defined in (2). Based on the results shown in the table, we conclude several implications as follows:
(1) The battery life for users who prefer one specific application category is difficult to increase. For example, users in type T_2 like *Phone Call* and the ratio r is only 1.03. We can not sacrifice the most preferred applications while others only have small influence on the device power.
(2) The *Sleep* time decides the battery time since the power in sleep state is too small compared with other applications. The ratio was over 1.6 in user type T_1 and T_5 as the corresponding time in sleep state increased to over 10%.
(3) The battery life can be extended up to 40% if users adjust application usage rather than put the device into sleep state. The difference of category power usually is around 500 mW. In some extreme cases, the difference can be 1000 mW. The time spent on each application category is less than 10% in most cases. Hence, if a user gives up a high power application and puts the saved time on another low power application, the difference of average power is about 100 mW. If the device average power decreases from 350 mW to 250 mW, the battery lifetime is 40% longer.

For mobile devices energy saving, the proposed strategies need to be more personalized. Users can set proper system and application configurations to extended battery life. For example, the pull option for email synchronization is more energy efficient than push for users who receive a lot of emails everyday. Besides, same energy optimization approaches will have different influences for different users. Users who like playing games will enjoy the energy savings on the GPU and CPU, while for the users who usually use the device to listen to the music probably will not notice the improvement.

B. *Hardware Component Improvement*

As more and more researchers work in the mobile devices energy saving field, a lot of optimization approaches are proposed for the system and various hardware components. We discussed the influence of the five main component improvements, which includes battery, CPU, radio, display and Wi-Fi, on the battery lifetime for each user type .

The hardware improvements correspond to the parameter s_i (and thus $\vec{M}' and \vec{A}'$) illustrated in the prediction model in Section II. For example, if the energy efficiency of CPU improves 10 times then s_1 is 1/10, and if the energy efficiency of radio improves 2 times, then s_3 is 1/2. Hence, the primary inputs are the power consumptions of each component for every kind of application and the improved energy efficiency of hardware components. We can use (2) to obtain the extended ratio of battery lifetime for each user type.

We logged the power consumptions of each component in previous experiments in Section III. For component improvement, the increased times of energy efficiency are demonstrated in Table VIII. According to Moore's law, the performance should be doubled in 18 months. So the improvement should be 2^3, nearly 10 times, in the next five years. However, they are not all energy proportional. *CPU*, as the most important component in mobile devices, is used by all applications, its energy efficiency may increase up to 10 times. Battery capacity is hard to increase as its developing history suggests, we assume it will increase 2 times in the future. For display, aside from the big improvement (10 times), the energy efficiency may also increase a little because its performance is acceptable and the appearance of new technology is hard to predict. We also considered two potential improvement for Wi-Fi as it supports the favorite function, wireless, of mobile devices. The characteristics of *radio* make it need to monitor phone calls since messages all the time and they are basic functions of a phone. We think its energy efficiency may be doubled in the future. Hence, we calculate the four cases listed in Table VIII that describe the trend of energy efficiency improvement for mobile device components in the next five years.

The remaining information in Table VIII presents the results of how many times the battery life can be extended for each user type in each of the four cases. The extended ratio grows linearly to the battery improvement. If the improvements of all the components are the same, the result is also a linear function which is not related to user behavior, but this situation is very rare. The results for case 1 and case 2 are around 5.5, while the results for case 3 and case 4 vary between 8 to 12 (except for users in T_3). The difference between the two groups is the display improvement. Because the average power of the display of all application categories is almost 40% of

TABLE VIII. LIFE TIME IMPROVEMENT FOR DIFFERENT USERS AND CASES

	Battery	CPU	Radio	Wi-Fi	Display	User Type T_1	User Type T_2	User Type T_3	User Type T_4	User Type T_5	User Type T_6
Case 1	2	10	2	2	2	5.95	5.73	4.26	6.26	5.76	5.35
Case 2	2	10	2	10	2	6.08	5.73	4.32	6.41	5.88	5.45
Case 3	2	10	2	2	10	10.05	8.95	5.54	11.26	9.38	8.11
Case 4	2	10	2	10	10	10.42	8.95	5.63	11.75	9.71	8.34

the device's power, its influence can double the extended ratio at most. For users in type T_3 who put their device in the sleep state for most time, the impact is not as great as in other cases. In case 4, T_4 users can enjoy 11.75 times battery life time extension, which is about 2 times more than T_3 users. The results show that user behavior can affect the battery life time a lot even for the same hardware improvement.

Next, we look at how much the hardware should be improved to achieve a target battery life, such as seven days, for different types of users. The average power of the main components for each application category is known as shown above. We can calculate the user power demand for each component according to the application usage. Suppose we want to improve the battery life by r times, the capacity of the battery can improve x_0 times, and denote

$$N = (\vec{M}_1'; \vec{M}_2'; ...; \vec{M}_n')$$

, then we can calculate how much should be improved by solving a linear equation

$$sum(N * \vec{B}^T) = \frac{x_0}{r}$$

Note that N is a matrix and since $\vec{M}_i'$ is just s_i times each elements in $\vec{M}_i$, so the variables in the linear equation are s_i. For example, if T_4 users want to enjoy a one week experience under the condition that the capacity of the battery can only improve 2 times, then we need to improve the energy efficiency of CPU by 7.6 times, display by 6 times, Wi-Fi by 1.45 times and radio by 3 times.

Generally speaking, the optimization approaches for components which are required by most applications have more impact, such as battery, CPU and display. For different users, the improvement of the "most used" components which are inferred from user behavior is more effective for them. Moreover, we discussed the theoretical influence of components' improvement, while the actual results should also consider the software interference.

V. DISCUSSION

The prediction model can calculate the battery life time with a relatively high accuracy, while there are still several points that are not considered. In this section, we discuss three aspects that may influence the prediction result.

Device Sleep Frequency: Users can set their preferred screen stay on time after there is no interaction detected. For the users who like pushing the on/off button to manually put devices in the sleep state, there is no display energy waste. So the sleep frequency does not affect the prediction model. However, for others that prefer to wait for the display to automatically turn off, the sleep frequency must be accounted. The high sleep frequency causes more energy waste. The power waiting display timeout is greater than the screen off device power, the battery lifetime will decrease. Hence, for the second type of users, we should add display timeout power in the model as a linear function of the device sleep frequency.

Foreground vs. Background: In the prediction model, the application usage behavior mainly focuses on the foreground applications and a few background applications (e.g. download applications). Modern mobile devices support multitasking, so background applications can refresh themselves and have some activities. Some background applications may cause the system to wake up from the sleep state or change the Wi-Fi from the idle state to the active state periodically. In our prediction model, we did not consider such background applications since there is no clear relationship that illustrates their influence on the battery life. These background applications may generate different power behaviors when the system configuration and other applications' states are different. Most of the background applications consume very little energy, however there are still some exceptions that require special attention.

User Behavior: In this paper, we use the percentage of application running time to distinguish different types of users. While it is only one of the aspects that describe user behavior, there are many other factors that are also included in user behavior. Besides, users' interaction with the same application is different. For example, some people prefer to read customer reviews rather than go through all the pictures when buying a product. Different functions of the application consume different energy. Our prediction model can handle this situation since it calculates the power directly from hardware resources. But it is difficult to classify users according to application functions. The monitoring overhead is huge. Hence, we leverage application running time to represent user behavior and apply an average functions' power as application power, which may affect the accuracy of the analysis results.

VI. RELATED WORK

To get the devices power information, the intuitive way is measuring devices by power meters. Previous research work [15] directly attaches power meters or sensors to device to analyze devices' power behavior under different workloads. In order to get power information more convenient, researchers calculate power through power models rather than hardware measurements. Zhang *et al.* [10] collected power traces for hardware components and built power meter based and battery based power models. Pathak *et al.* [16] proposed system-call-based fine-grained power estimation by modeling power state using Finite State Machines (FSM). Generally speaking, these power models are built according to the different power states of hardware components, the trigger of the state change and the approach to get the trigger information are distinct. In our paper, we adopt hardware utilization based power models and the trigger is the component usage information. Its error is within 10% and the overhead is less than system call or kernel activity monitoring. Moreover, it helps us build the relationship of the battery extended time and hardware improvement more easily and conceivable.

As mobile devices are becoming more and more popular, the user interaction with the devices gains researchers' attention. Some previous works [7], [17] demonstrate the user behavior information which includes interactive duration, top applications and their usage, network packets and so on. The results showed the user diversity in most of the metrics they studied. Hence, it gives the space for us to extend battery time by changing user behavior.

Battery drain is an important problem, and it is necessary to build an accurate battery lifetime prediction model so that optimization methods can be applied based on the battery capacity. Zhao *et al.* [18] proposed a system context-aware approach which calculates battery discharge rate from the states of system components. Ravi *et al.* [19] estimated battery time according to the battery discharging curve and took battery charging behavior and phone calls into consideration. Kang *et al.* [20] presented a personalized battery lifetime prediction method based on usage patterns (sleeping, working, etc.). Our model is more general and illustrates the relationship of battery life with hardware components and application usage.

VII. Conclusions and Future Work

Battery drain is a primary issue in mobile devices. Our prediction model estimates the battery lifetime based on user behavior and power consumption information from hardware components. We did several experiments to verify the assumption that application power is relatively stable. Since the application power is calculated from power models, we analyzed their influence on the accuracy of the prediction model. In order to analyze the influence of user behavior, we classified users into six groups according to their application usage pattern and estimated the theoretical maximum battery life time they can achieve. We presented the potential time that battery life can be extended under different improvements of the main hardware components, which gives us a basic idea of battery life of future mobile devices. Generally speaking, we analyzed how much work needs to be done if we want to extend battery life time to reach a certain goal, like seven days, from user behavior and hardware components aspects. In the future, we will improve the accuracy of the prediction model and propose some user behavior optimization strategies to help users extend battery life. Moreover, we will pay attention to software improvement which also affects battery lifetime.

Acknowledgment

The authors would like to thank Charles Guye for his help on proofreading and the anonymous reviewers for their comments and kindly suggestions.

References

[1] J. Cosley, "In a mobile world, not all mobile performs equally for advertisers," Apr 2014, http://marketingland.com/mobile-world-mobile-equal-78561.

[2] P. Daniel, "Battery life to be the single main gripe of today's mobile phone user," http://www.phonearena.com/news/Survey-shows-battery-life-to-be-the-single-main-gripe-of-todays-mobile-phone-user_id49818, Nov 2013.

[3] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, "Ecosystem: managing energy as a first class operating system resource," *SIGPLAN Not.*, vol. 37, no. 10, pp. 123–132, 2002.

[4] A. Roy, S. M. Rumble, R. Stutsman, P. Levis, D. Mazières, and N. Zeldovich, "Energy management in mobile devices with the cinder operating system," in *Proceedings of the sixth conference on Computer systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 139–152. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966459

[5] R. Duan, M. Bi, and C. Gniady, "Exploring memory energy optimizations in smartphones," in *Proceedings of the 2011 International Green Computing Conference and Workshops*, ser. IGCC '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/IGCC.2011.6008591

[6] J. Manweiler and R. R. Choudhury, "Avoiding the rush hours: Wifi energy management via traffic isolation," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 253–266. [Online]. Available: http://doi.acm.org/10.1145/1999995.2000020

[7] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 179–194. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814453

[8] H. Chen, Y. Li, and W. Shi, "Fine-grained power management using process-level profiling," in *Sustainable Computing: Informatics and Systems*, ser. SUSCOM, vol. 2, no. 1, Mar 2012, pp. 33–42.

[9] BP Precision, http://www.bkprecision.com/, model 1785B.

[10] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "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*, ser. CODES/ISSS '10. New York, NY, USA: ACM, 2010, pp. 105–114. [Online]. Available: http://doi.acm.org/10.1145/1878961.1878982

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

[12] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[13] D. Raychev, Y. Li, and W. Shi, "The seventh cell of a six-cell battery," in *third Workshop on Energy-Efficient Design, WEED*, Jun 2011.

[14] Google, "Nexus 4 technique specifications," Nov 2012, https://support.google.com/nexus/answer/2840740?hl=en.

[15] J. Flinn and M. Satyanarayanan, "Powerscope: A tool for profiling the energy usage of mobile applications," in *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 1999.

[16] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 153–168. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966460

[17] A. Rahmati and L. Zhong, "Studying smartphone usage: Lessons from a four-month field study," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 7, pp. 1417–1427, July 2013.

[18] X. Zhao, Y. Guo, Q. Feng, and X. Chen, "A system context-aware approach for battery lifetime prediction in smart phones," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 641–646. [Online]. Available: http://doi.acm.org/10.1145/1982185.1982327

[19] N. Ravi, J. Scott, L. Han, and L. Iftode, "Context-aware battery management for mobile phones," in *Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, ser. PERCOM '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 224–233. [Online]. Available: http://dx.doi.org/10.1109/PERCOM.2008.108

[20] J.-M. Kang, S.-S. Seo, and J. W.-K. Hong, "Personalized battery lifetime prediction for mobile devices based on usage patterns," *JCSE*, vol. 5, no. 4, pp. 338–345, 2011.