Personal and Ubiquitous Computing (2022) 26:821–836
https://doi.org/10.1007/s00779-019-01213-4

ORIGINAL ARTICLE

A framework for usage pattern–based power optimization and battery lifetime prediction in smartphones

Nirmal Pandey [1] · Om Prakash Verma [1] · Amioy Kumar [1]

Received: 22 March 2017 / Accepted: 14 March 2019 / Published online: 13 April 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract
The use of mobile devices has increased many folds over the last few years. Smart phones are used not only for communication but also for storage of personal contents like photos, videos, documents, and bank account and credit/debit card details. Secure access to these contents is very important. Nowadays, many mobile devices come with inbuilt biometric-enabled security features like fingerprint, face recognition, and iris. However, additional battery power is consumed each time a user unlocks the device using any one of these security features. In order to enable prolonged use of the device, there is a strong need to find ways to conserve power in mobile devices. At the same time, it is also equally important that the smart phone user knows how long the battery of his device will last. In this paper, we present a novel power optimization and battery lifetime prediction framework called P4O (*P*attern, *P*rofiling, *P*rediction, and *P*ower *O*ptimization). Our contributions are threefold—(i) Propose a novel framework for power optimization in smart phones. (ii) Propose a new approach for battery lifetime forecast. (iii) Implement and validate the efficacy of the proposed framework. For experimental results, the proposed framework was implemented on the Android-based smartphone. The experimental results validate the proposed framework with power optimization up to 40% over default Linux and Android power saving features available in an Android operating system. This framework is also able to forecast battery lifetime with accuracy of up to 98%.

Keywords Energy efficiency · Mobile devices · Battery lifetime · Usage pattern · Power optimization

1 Introduction

The adoption of smart phones in society is increasing at a very fast pace. As per one estimate, about 432 million units of smart phones were sold worldwide in the last quarter of 2016 [1]. Till recently, patterns, personal identification numbers, passphrases, and passwords were the most common ways to unlock smart phones. According to a recent survey [2], one of the top concerns of the smart phone users was related to the security of the device. Due to these concerns, smart phone OEMs are enhancing the security features of mobile devices by adding biometric-enabled security features such as fingerprint scanners, face, and iris recognition [3, 4].

For the purpose of testing the power consumed by biometric components in smart phones, we used the Samsung Galaxy S7 Edge [4] mobile phone as the device under test. It has an inbuilt front-facing fingerprint scanner. For using the face recognition on Galaxy S7 Edge, we have used a third party app—the IOBit Applock [5]—for power measurement. As we observe the Galaxy S7 Edge power readings for a fingerprint and face unlock as given in Fig. 1, we see that for each unlock operation, 200 to 400 mA extra current is consumed. When added with the fact that on an average, one smart phone user unlocks his device 100 times daily [6], a total power consumed due to biometric sensors could range from 5 to 10% of total battery capacity in a day. Thus, biometric-enabled security features reduce the number of hours a smartphone can be used without charging. This reflects the severity of the power consumption problem in smart phones. This has been corroborated in many surveys and studies carried out in the past. Rahmati et al. [7] found that 80% of the mobile users

✉ Nirmal Pandey
writetonirmal@gmail.com

Om Prakash Verma
opverma.dce@gmail.com

Amioy Kumar
amioy.iitd@gmail.com

1 Delhi Technological University (DTU), Shahbad Daulatpur, Main Bawana Road, Delhi 110042, India

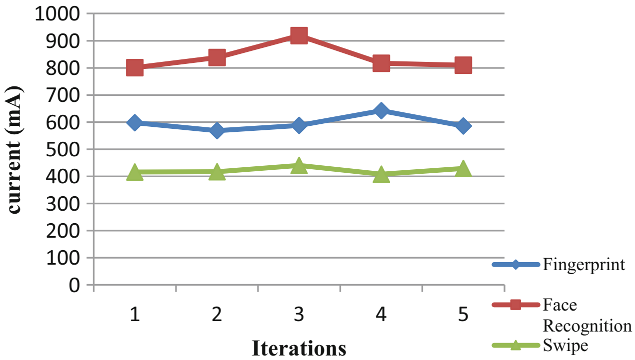


Fig. 1 Current consumption for fingerprint and face unlock features

were concerned with the battery life of the device. Therefore, a systematic and comprehensive approach is needed to overcome power consumption and fast battery drainage problem in smart phones.

A large number of studies have been carried out by many researchers to find out power consumption characteristics in mobile devices. Rahul Balani [8] analyzed energy consumption for various radio transmission technologies such as BT, Wi-Fi, 2G, and 3G. Andrew Rice et al. [9] developed a measurement framework that logs the traces of the consumed power. This was done to help in understanding the power consumption from the application perspective. Aaron Carroll et al. [10] analyzed and broke down power distribution into various components such as processors, display, and primary and secondary storage, and also developed an overall energy model of the device.

G.P. Perrucci et al. [11] studied energy consumption entities on the smart phone platform and identified various energy-hungry components of the mobile phones. Paul et al. [12] studied power consumption in Android devices. They proposed that more energy is consumed in the Android Dalvik VM (virtual machine) due to the lack of dynamic compiler in comparison with the Sun Embedded JVM (Java virtual machine).

By closely observing the power consumption profile of a mobile device, we find that some components of the device consume more power than others. Major power consuming components are CPU, GPU, display, radios, biometric sensors, other sensors (accelerometer, proximity, gyro, etc.), and audio components. During the unlocking of smartphones, the power consumed by biometric sensors is common in all use cases. Some or all of these components are active in various use cases of the device. Here are a few common instances:

1. Playing a video stored locally on the device (internal or storage card) involves unlocking the device (biometric sensors), CPU, GPU, display, speaker and storage card.
2. Playing a video from YouTube or any other streaming server involves unlocking the device (biometric sensors), radios, CPU, GPU, speaker, and display.
3. Making a voice call/video call involves unlocking the device (biometric sensors), radios, CPU, display (some part), sensors (proximity), camera, and speakers.
4. Playing a 2D/3D game involves unlocking the device (biometric sensors), CPU, GPU, display, speaker, and various other sensors.
5. Internet browsing involves unlocking the device (biometric sensors), radios, CPU, GPU, and display.

From these use case scenarios, it is obvious that any meaningful use of the smart phone involves a large number of software and hardware components. All of these components consume power. So, any power optimization system should encompass power saving in all or most of these components in order to be useful. The smartphones have many inbuilt power saving features available in the device. Figures 2 and 3 below show some of the power saving features available in the Samsung Galaxy S7 Edge [4] Android smart phone.

As we can see from the Figs. 2 and 3 above, in order to carry out effective power saving, a user must be technically conversant with power saving terminology such as screen brightness, screen resolution, CPU processing speeds, and network usage. along with the relationship between these parameters and corresponding power consumption. This is not the case with most of the users [7]. Therefore, an effective power optimization framework that does not require the user to fully understand power consumption domain is needed. To address this problem, a new framework is proposed in this work (detailed in the "Contributions" section). To optimize power and predict battery lifetime, a novel framework called P4O (*P*attern, *P*rofiling, *P*rediction, and *P*ower *O*ptimization) is proposed and investigated in detail in this work (detailed in the "P4O framework" section).

The rest of the paper is organized as below. In the "State of the art and contributions" section, we present a survey of prior research done related to the usage pattern–based mobile phone power consumption and battery lifetime prediction. The proposed power optimization framework P4O is presented in the "P4O framework" section. In the "Power consumption pattern and power states" section, we have covered the power consumption pattern and power states. The experimental results are provided under the "Experimental results" section. Under the "Conclusion and future work" section, the conclusion of the work done is given and the scope for future research is outlined.

2 State of the art and contributions

2.1 Prior work

Power optimization and accurate battery lifetime prediction problems have drawn the attention of many researchers to this

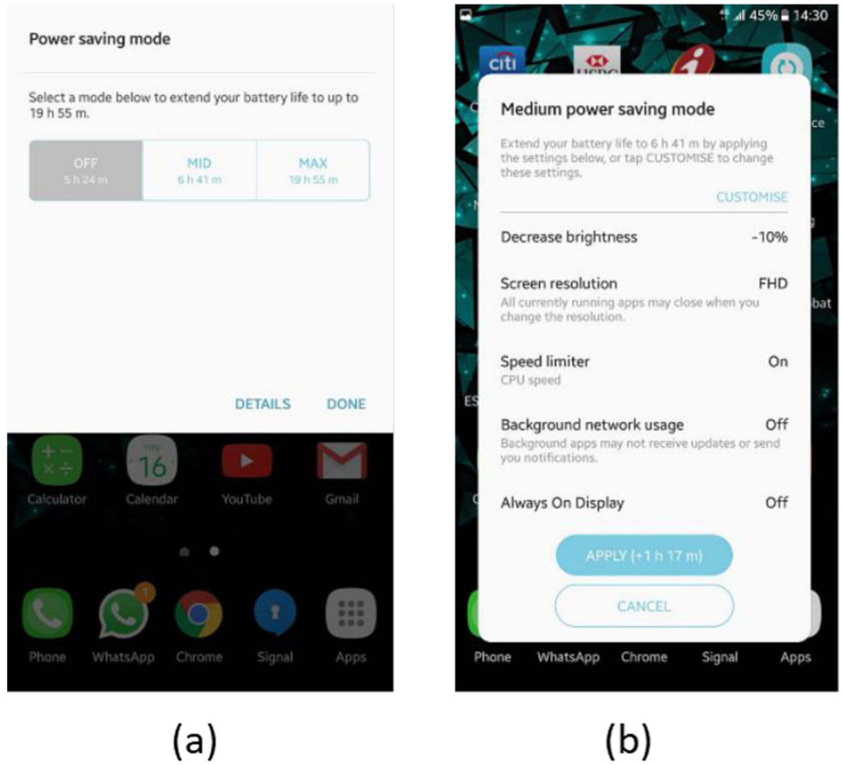


Fig. 2 Samsung Galaxy S7 Edge Power Saving. **a** Modes. **b** Medium power saving parameters

field. Analysis of user behavior as well as device usage patterns has been investigated by various researchers. Lyons et al. [13] conducted a 4-week investigation involving more than 4000 users, but they focused on the charging habits of smartphone users. R. Demumieux et al. [14] made use of various functions available on the device and collected the information along with the time taken for each of the activities. Rahmati et al. [7] studied the human-battery interaction of

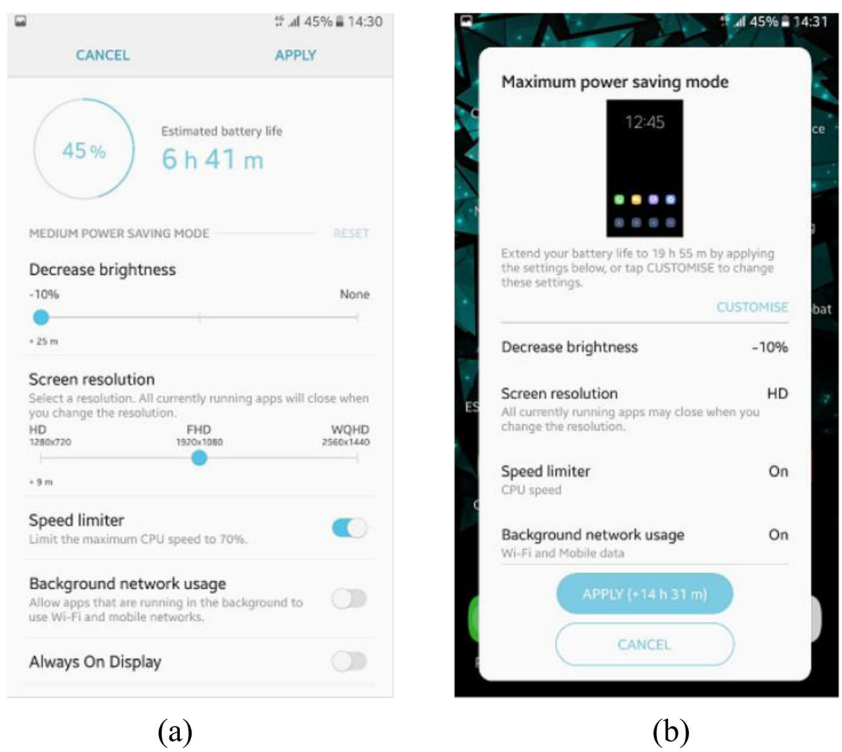


Fig. 3 **a**–**b** Samsung Galaxy S7 Edge maximum power saving mode

mobile phone users. They also studied GUI-related aspects of battery information in mobile phones. Kang et al. [15] analyzed the usage pattern of mobile phones. They took phone operations such as voice call into consideration and defined the various states of the device to understand user behavior. E. Oliver et al. [16] presented the challenges in conducting large-scale user studies. They examined user mobile interactions and energy consumption of the device. J. Lee et al. [17] proposed an automated power model generation approach taking usage patterns into consideration. The usage patterns were put under various baskets and tested for power model generation. H. Falaki et al. [18] focused on collating logs and presented a diversified usage pattern. Verkasalo [19] presented a framework called MobiTrack for the data collection and analysis of user behavior. Oulasvirta et al. [20] examined various data sources on the mobile device and concluded that mobile users form habits of device usage. Arun et al. attempted to provide an innovative plan for effective services to end users on the deployment of information services through mobile apps [21]. Alexander W. Min et al. [22] presented a framework called E2S3 (energy-efficient sleep-state selection). This framework minimizes battery consumption by adaptively going into an optimal low-power sleep state. In the direction of building IOT (Internet-of-things) sensors, a work done by Turjman [23] and Sing et al. [24] is also worth mentioning. The work in [24] attempted on information-gathering networks in large-scale IOT applications such as smart cities, which serve multiple users with diverse quality-of-information (QoI) requirements on the data delivered by the network.

Battery lifetime prediction has also been investigated by various researchers. J.M. Kang et al. [25] presented a battery lifetime prediction method that makes use of activities and the time spent in these activities as the basis for lifetime computation. Chandra et al. [26] presented application-level battery consumption. However, they did not consider the diversity of usage by various users. Zhang et al. [27] proposed an automated power model construction method called PowerBooter. They also proposed a power management tool called PowerTutor that works in conjunction with PowerBooter.

Various researchers have investigated the smartphone applications usage pattern. Some of them have investigated app usage to relate it to the stress level of the smartphone user. Research has also been done to investigate the app usage pattern based on the GPS location of the user, based on the wearable and sensor use in the smartphone. However, none of these investigations are focused on the relationship between the device usage pattern and battery drop rate, thus paving a way for opportunistic power optimization in the device.

A comparative study on some recent state of the art and the present work are shown in Table 1. It can be noted that the power optimization in smart phones and battery life forecast are a relevantly new area of research and carried out in none of the popularly available state of the arts.

2.2 Motivations

Every mobile user has a consistent and unique usage pattern [33]. Therefore, any power optimization technique also needs to be unique to each individual. Generic power conservation methods will have limited success as each individual differs in the way he interacts and makes use of his device. While capturing user behavior patterns, it is equally important that the data captured does not violate the privacy of individuals. Many usage pattern–based methods proposed in the literature make use of sensitive information such as user location, which applications user accesses and for how long, etc. These raise serious privacy issues, if such information is shared to web servers for processing. In addition to this, client-server-based power optimization architectures suffer from the lacunae that these are not real time, rather near real time. This affects the accuracy of these systems.

Some of the battery lifetime prediction methods available in the literature use static information for prediction, which gives incorrect output. Most other techniques available in the literature make use of the application usage pattern benchmarking. This also has the limitation as most users frequently change old applications and install new applications on the device.

2.3 Contributions

In this paper, our main contributions are as follows:

1. A novel framework for power optimization referred to as P4O (*P*attern, *P*rofiling, *P*rediction, and *P*ower *O*ptimization) is proposed. The proposed framework can learn the battery usage pattern of a user over a period of time. It utilizes the Kalman filter to provide knowledge base updates. This can optimize the battery power based on the state transition (see the "Power consumption pattern and power states" section for detail). Based on the learned pattern, it can minimize battery usage and hence optimize the battery power.
2. This framework has many advantages. It solely runs on the device. Therefore, there is no security or privacy aspects involved in the use of this framework. It makes use of the battery discharge rate information for benchmarking user behavior. This makes it independent of applications users tend to use, as users tend to install/uninstall applications.
3. It proposes a new approach for battery lifetime forecast. The proposed method makes an accurate prediction of battery lifetime based on the usage pattern analysis.
4. The proposed framework has been implemented on Android [34] smart phone. Its efficacy has been validated in experimental test results.

Table 1 State of the art in usage pattern analysis

S. no.	Ref	Application	Research methodology	Accuracy	Optimization
1	Ferdous et al. [28]	Smartphone app usage as a predictor of perceived stress levels at the workplace	- Studied the patterns of app usage, specifically the types of apps and their duration. - Proposed the relationship between the app usage pattern and the stress level of the user.	Average accuracy = 75% Precision = 85.7%.	Not available
2	Turjman et al. [29]	Ubiquitous cloud-based monitoring via a mobile app in smartphones: an overview	- Investigated the impact of wearable devices on user habits. - Focused on two areas—usage patterns and user mobility tracking using smartphones.	Not available	Not available
3	Elgedawy et al. [30]	IdProF: Identity provisioning framework for smart environments	- Proposed a secure context-sensitive seamless multi-modal identity provisioning framework for smart environments. - Can generate a disposable customized virtual inhabitant profile (DCVIP) on demand. - Creates a customized identity proxy to handle the identity verification for the required interaction.	Not available	Not available
4	Li et al. [31]	Characterizing smartphone usage patterns from millions of Android users	- Presented an empirical analysis of app usage behaviors collected from millions of users of Wandoujia, a leading Android app marketplace in China.	Not available	Not available
5	Lu et al. [32]	Mining mobile application sequential patterns for usage prediction	- Mining and prediction of mobile application usage behaviors. - Proposed a location-based approach to predict the mobile application usage behaviors.	Precision = 32% Recall = 40% (Both at 3% threshold)	Not available
6	This work	Novel power optimization and battery lifetime prediction framework	- A novel framework for power optimization in smart phones. - A new approach for battery lifetime forecast	Accuracy = 73% (With 2 weeks' usage pattern knowledge)	Available (Power optimization up to 30% over default Linux and Android systems)

3 P4O framework

The proposed P4O framework intends to record, analyze, learn, predict, and optimize the battery power consumption of the smart phone in a non-invasive way. Smartphone users exhibit a pattern of usage. This usage pattern is also reflected in the battery discharge pattern of the device as well. The P4O framework proposes to record the battery discharge pattern for each usage pattern slot of 30 min each. The usage pattern, so learned, will be used to predict the power state and take the device to a lower power state whenever the opportunity arises. It will also predict the battery lifetime of the device as per the learned usage pattern. The P4O framework is explained below.

The main components of the proposed framework are the following: logging and learning module, prediction module, decision-making module, power state controller, and battery lifetime forecast module. Figure 4 depicts the proposed P4O framework.

3.1 Logging and learning module

The purpose of logging and learning module is to monitor the battery consumption statistics of the device and record it in the usage pattern DB. It analyzes and learns the battery consumption pattern of the device. It takes inputs from battery monitoring service that is running on the device and monitors battery drop/charge status. It also takes into consideration the time of the day supplied by the system clock for pattern analysis and learning the usage pattern. Battery drop/charge rates are recorded by this module into the battery DB for the time slots that are pre-defined.

3.2 Prediction module

The prediction module takes inputs from the logging and learning module as well as the system clock. It predicts the most suitable power state for the system for the current usage time slot. For a given usage time slot, logging and learning module gives information about the past usage behavior of the device. The prediction module forecasts the power state by taking this information into consideration.

3.3 Decision-making module

The purpose of decision-making module is to decide whether the power state of the device for the current usage slot needs to be changed. For this purpose, it takes the predicted power state

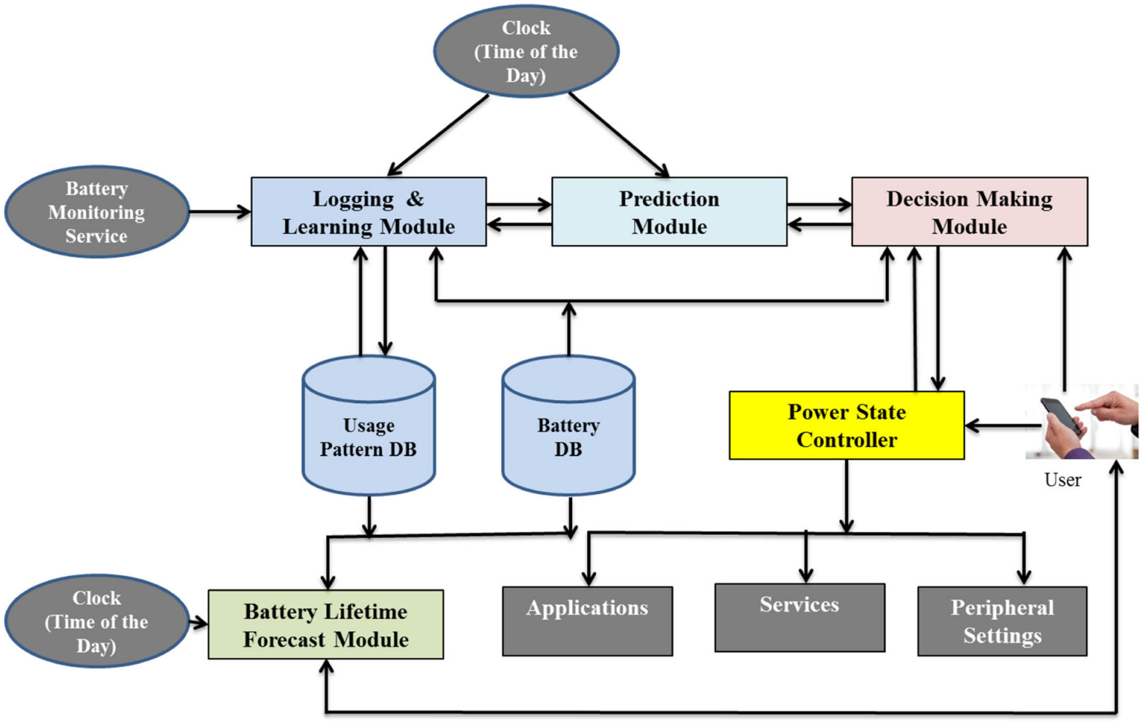


Fig. 4 P4O framework block diagram

from the prediction module, current user action, and time left in this usage slot as input, and decides the power state transition.

3.4 Power state controller

The P4O framework utilizes four power proposed states—high (H), medium (M), low (L), and very low (VL). These power states have been defined by taking the rate of battery discharge into consideration. If, for a given usage pattern slot, the battery drop rate is less than the computed threshold (refer to Table 2) but more than zero, the power state is declared a very low (VL) state. Similarly, other power states, low, medium, and high, are defined. The exact values of battery drop rates for these power states may vary from user to user. For example, if someone uses the device more for multimedia applications such as video streaming or high-end gaming, the value of battery discharge rate per usage pattern slot will be much higher than that for the person who uses the devices mostly for voice calls or sending messages. The power state controller takes the decision made by the decision-making module as input and does the actual work for making a transition from a higher power consumption state to a lower power consumption state. This module interacts with running applications, services, and various peripherals for actual power state transitions. However, if the system is woken up due to an external trigger, the power state controller rolls back the changes and takes the system from a lower power consumption state to a higher power consumption state depending upon the need of the system.

Table 2 Battery consumption rate to power state mapping

Power state	Battery consumption rate
Very low (VL)	*DR*
Low (L)	> *DR* and < (5 × *DR*)
Medium (M)	> (5 × *DR*) and < (10 × *DR*)
High (H)	> (10 × *DR*)

3.5 Battery lifetime forecast module

The purpose of battery lifetime forecast module is to predict the duration up to which the battery of the device will last or, after a given duration, how much battery will be left if there is no charging opportunity. This module takes the usage pattern of the device and the time duration from the user, and calculates the duration or battery lifetime.

4 Power consumption pattern and power states

The device usage pattern is reflected in the battery consumption or battery discharge pattern of the device. Therefore, recording this behavior becomes very important for any useful prediction and saving device power.

4.1 Building usage pattern knowledge base

For the purpose of recording the battery discharge pattern of the device, the period of interest is taken as 1 week. Each day of the week is further divided into "usage pattern slots" of 30 min each. Thus, we have 48 ($24 \times 2 = 48$) usage pattern slots for each day of the week. Initially, since there is no knowledge base of battery discharge pattern, the P4O framework records the battery discharge rates for each of these slots ($48 \times 7 = 336$ slots) for the entire week. This recorded information serves as the initial knowledge base of the usage pattern. All this information is recorded in real time in the device without user intervention. The flowchart below explains the building of the usage pattern knowledge base (Fig. 5).

4.2 Battery discharge pattern knowledge base update

The battery discharge rate information for each of the usage pattern slots is captured in real time. Even though the way any particular person uses the device shows some pattern, it may still change. Thus, the knowledge base, which stores the previous record of battery discharge rate for each of the 336 usage pattern slots, also needs to be updated in real time. In our system, we used the alpha-beta filter to calculate the updated battery discharge rate by taking the old value and the new value of the battery discharge rate.

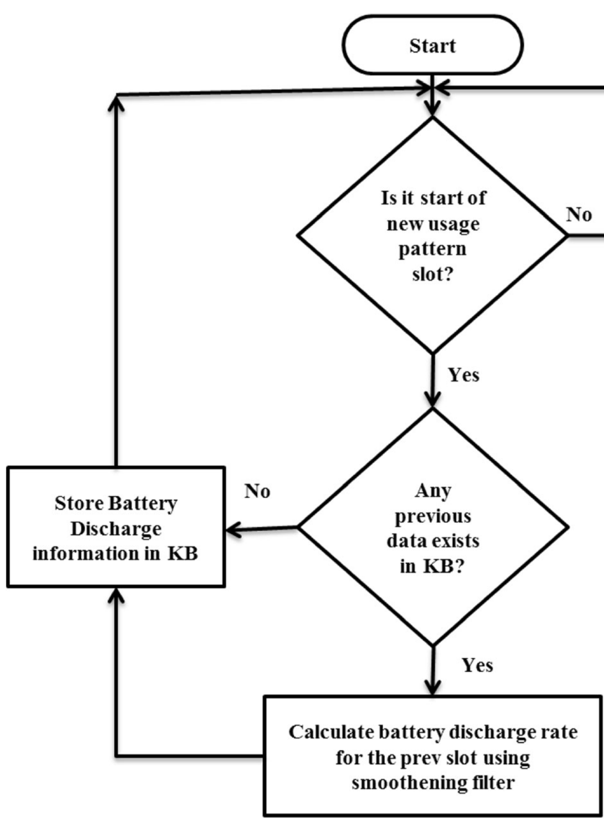


Fig. 5 Building battery discharge usage pattern knowledge base

The alpha-beta filter is a simpler form of the Kalman filter [35]. Let us assume that d_k is the battery discharge rate of the kth usage pattern slot and d_{k+1} is of the same slot during the same day of the next week. In that case, the new value of battery discharge rate for $k+1$st slot is given by the following equation:

$$d_{k+1} = d_k + \alpha \left(d_{k+1} - d_k\right) \tag{1}$$

Here, α is the smoothening factor between 0 and 1. A very high value of alpha means the system will catch up with the changes very fast, while a value of zero means the system will ignore any changes reported in the system. In the P4O framework, the value of alpha is kept at 0.5.

4.3 Power consumption states mapping

P4O framework defines four power consumption states. These states are dependent upon the value of battery discharge rates. The four power consumption states are high (H), medium (M), low (L), and very low (VL). The following algorithm is used to map the battery discharge rate to power states. This algorithm helps in creating the mapping between the battery consumption rate and the power state independent of battery capacity. The steps mentioned below are used in the algorithm:

1. Find the total battery capacity (T_{B}) of the device [in milliampere hours (mAH) units].
2. Assume the device base current (I_{B}). For Android smart phones, the base current is considered 10 mA in experiments after measurements [36].
3. Calculate the idle usage time of the device (T_{U}) [in hours unit]:

$$T_{\mathrm{U}} = T_{\mathrm{B}} / I_{\mathrm{B}} \tag{2}$$

4. Calculate the battery discharge rate for one usage pattern slot of 30 min:

$$\text{Battery drop rate per hour} = 100 / T_{\mathrm{U}}$$

Therefore, the battery drop rate for 30 min will be given by the equation:

$$D_{\mathrm{R}} = \left(0.5 * 100 / T_{\mathrm{U}}\right) \tag{3}$$

5. The value of D_{R} computed from Eq. 3 is mapped to the power state "Very Low" as this is the minimum current that the device will draw even when there is no load on the device.

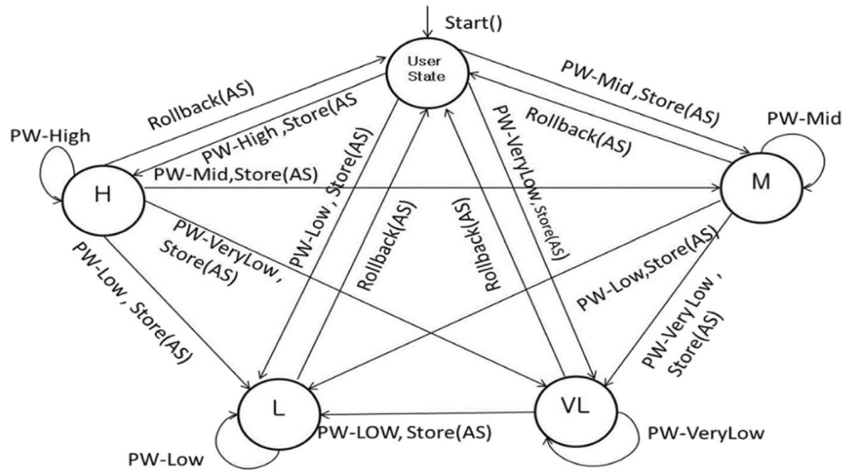


Fig. 6 Power state transitions

Table 3 Power state transition actions

Function	Very low	Low	Medium	High	Function	Very low	Low	Medium	High
Wi-Fi	X	X	O	O	Location/GPS	X	X	O	O
Bluetooth	X	X	X	O	CPU power Saving	X	X	X	O
Wi-Fi direct	X	X	X	O	Motion and gesture	X	X	O	O
Wi-Fi hotspot	X	X	X	O	Smart screen functions	X	X	X	O
Auto-sync	X	X	O	O	Tool box	X	X	X	O
Download booster	X	X	X	O	Bike mode/car mode	X	X	O	O
Smart bonding	O	X	X	X	Smart view	X	X	X	O
NFC	X	X	X	O	Glove mode	X	X	O	O
Brightness	X	X	O	O	Smart pause	X	X	X	O
Touch sensitivity	X	X	O	O	Smart stay	X	O	O	O
Screen rotation	X	X	O	O	Hands-free mode	X	O	O	O
Screen timeout	X	O	O	O	Reading mode	X	X	O	O
Screen power saving	X	X	X	O	S preview	X	O	O	O
Edge screen	X	X	X	O	Voice control	X	O	O	O
Key light duration	X	X	O	O	Air gesture	X	X	O	O
LED indicator	X	O	O	O	Air view	X	O	O	O
Battery level indicator	X	O	O	O	Multi window	X	X	O	O
Easy mode	X	X	X	O	Hands-free operation	X	O	O	O
Auto-rotate	X	X	O	O	Blocking mode	X	O	O	O
Volume intensity	X	O	O	O	Network restriction	X	O	O	O
Vibration intensity	X	X	O	O	Talk back	X	X	O	O
Haptic feedback	X	X	O	O	S pen	X	X	O	O
Volume	X	O	O	O	S cover	X	X	O	O

6. The remaining power states are mapped to the battery discharge rate as per Table 2.

4.4 Power consumption states transitions

The P4O framework is constantly on the lookout for conserving power. Power state transitions (higher power consumption state to lower power consumption state or vice versa) happen at either the start of a new usage pattern slot or when the system is woken up/goes to sleep due to user action (e.g., screen on/off) or some external trigger (e.g., incoming voice call/call disconnect). Figure 6 depicts the power state transitions. Each transition has a label which encapsulates various actions done by the P4O framework in carrying out that particular transition.

Some of the possible actions to be taken when changing power states are given in Table 3.

4.5 Battery lifetime forecast

As mentioned earlier, the logging and learning module of the P4O framework records the battery discharge rates for each of 336 usage pattern slots. It also updates the discharge rate values whenever there is a deviation from the previously stored value using a smoothening filter. Since this is recorded in real time as and when the mobile phone user uses the device, this information can be used to accurately compute the battery lifetime. The P4O framework also takes into consideration the user charging behavior of the device while computing the battery lifetime. While recording the battery discharge rate information for any of the usage pattern slots, if the device is undercharging, the battery discharge rate for that slot is considered zero. Thus, the recorded battery discharge rate information inherently incorporates the user device charging pattern as well.

Mathematically, we can represent this problem as follows. Let D_i denote battery discharge rate (in percentage terms) corresponding to usage pattern slot i. Let t represent the duration of the usage pattern slot, which is 0.5 h in our case. Let B_C be current battery level (in percentage terms). Battery lifetime is computed as per below flowchart (Fig. 7).

5 Experimental results

The P4O framework was implemented on a Samsung Galaxy J2 smartphone [37] running Android Marshmallow. Four volunteers were chosen, and the software of their devices was updated with the P4O framework–related changes. The volunteers were asked to use their devices normally as they would have used otherwise. The experimental information was extracted

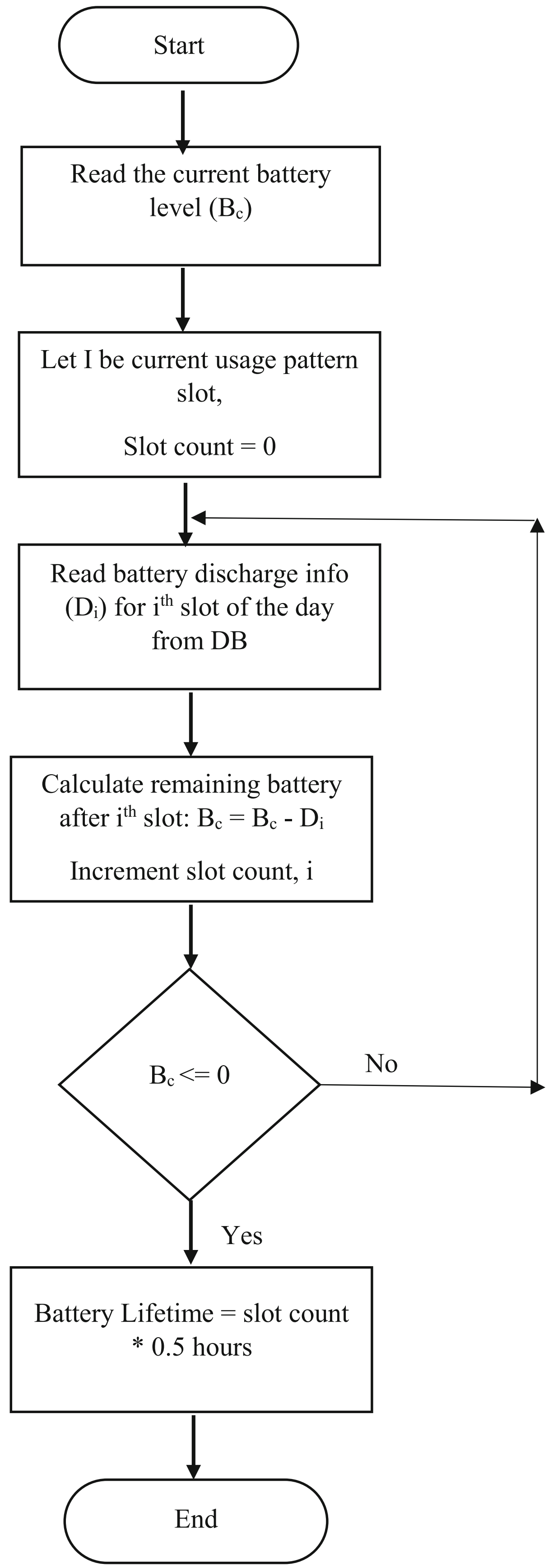


Fig. 7 Battery lifetime prediction algorithm

from these test devices every day. The overall results are grouped into four main sections: (1) battery power consumption pattern learning—here, normal devices (without P4O framework) are assigned to volunteers for usage pattern learning, (2) power state prediction of the learned patterns—here, the P4O framework–enabled devices are given to volunteers to test and provide the statistics on the power state prediction of the P4O framework, (3) battery power optimization as per predicted patterns—the P4O framework is tested for battery optimization in the real-time usage, and (4) battery lifetime forecast as per optimized battery power for user convenience. The outcomes from the abovementioned experiments are given in the below sections.

5.1 Battery consumption pattern learning

The purpose of this set of experiments is to train the device against the usage pattern data of the user. We selected 10 volunteers to do real-time experiments on the Samsung Galaxy J2 devices *without having the P4O framework*. We took volunteers from the staff group of Delhi Technological University, Delhi, India. It is important to note that these sets of experiments require a longer period of observation (at least 2 weeks) for reliable results. Hence, it requires a supportive set of volunteers who can judiciously use the device to provide trustworthy learning to the device in order to get reliable results at prediction. The usage pattern data of 10 volunteer test devices was observed for a period of 2 weeks. The 24 h of a day is divided into 48 windows of 30 min each, and the data of battery consumption is captured for each window. *These windows are referred to as slots from here onwards*. In each of these slots, the transition of the four states (high, medium, low, and very low) is marked (see the "Power consumption states mapping" and the "Power consumption states transitions" sections) and the battery data discharge rate is computed (See Table 2). This data (battery drop/consumed) is captured continuously and stored into a buffer until a broadcast of the shutdown, charging, or date change occurs. Whenever these changes occur or the buffer is full, the captured data is saved in a database in order to prevent any loss of the captured data.

In order to show that the battery usage by each user follows a pattern, we investigated the usage pattern slots (the "Building usage pattern knowledge base" section) for each user. We divided each day into 48 slots and hence each user has 336 slots to capture usage patterns in a week. We compared these 336 slots for each user for two consecutive weeks. The formula for calculating the matching percentage is given below:

$$\text{Slots matching percentage} = (\text{number of matching slots} \times 100/\text{total slots}) \quad (4)$$

For each of the 10 users, the slot-matching information is depicted in Fig. 8. It can be observed from the figure that 84 to 94% of the slots have matching battery discharge rates.

A bar graph for each of the 10 users is shown in Fig. 9 which shows the matching and non-matching slot scores for each user. The higher number of matching slots for each user shows that the battery discharge rates over a period of time follow a pattern.

From Figs. 8 and 9 above, we can conclude that the battery discharge rates over a period of time follow a pattern. This validates our assumption that every user exhibits a particular usage pattern which, in power consumption scenario, is reflected in the battery discharge pattern of the device.

5.2 Power state prediction

Based on the pattern learned in the above section, the next task is to test if the proposed P4O framework is able to predict the learned battery power consumption states in real time. This set

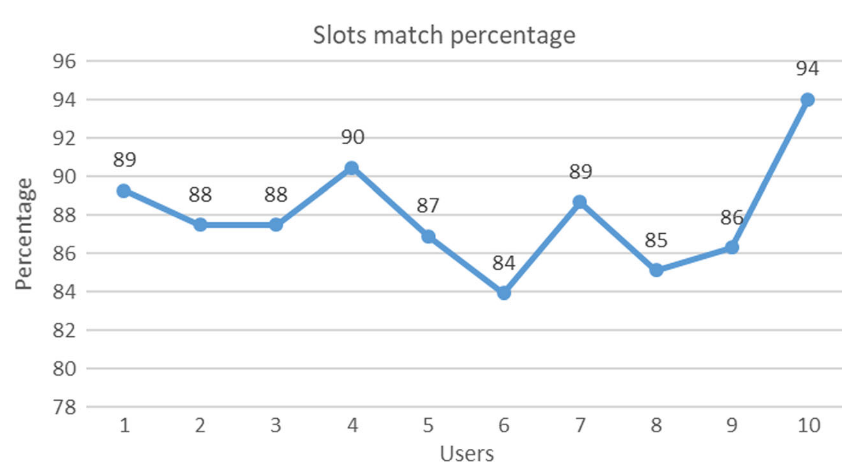


Fig. 8 Battery discharge slots match pattern (10 users)

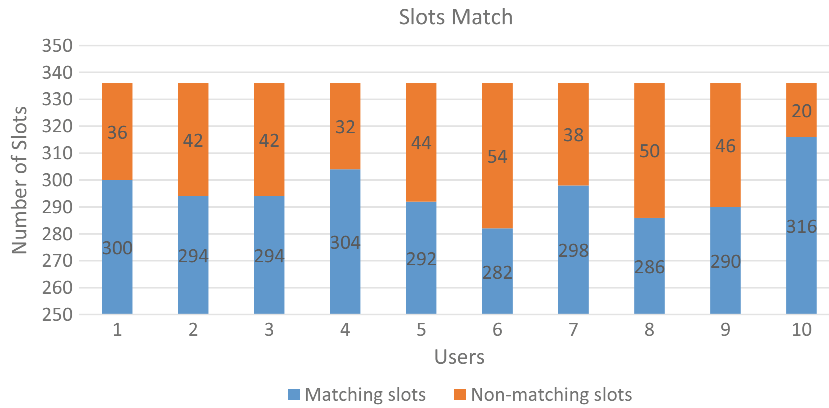


Fig. 9 Slots matching score (10 users)

of experiments is designed to evaluate the prediction capability of the P4O framework for the new time slots for the same user. *The same users have given the same devices (Samsung Galaxy J2) enabled with the P4O framework for 2 weeks*. The data collected with the predicted (P4O framework enabled) device is then compared with the actual (without P4O framework) device. Figure 10 depicts the comparison of the predicted versus the actual power state of one user for 1 day. It can be noted that the P4O framework is able to correctly predict the learned pattern of the power state for most of the usage pattern slots. The data depicted is taken for two consecutive weeks for experimental validation. However, with more learning or continuous learning mechanisms, the accuracy of the system would increase.

The accuracy and other statistics of the proposed P4O framework can be derived from the experimental results. As discussed in the previous section, there are 48 slots for each user and there are four users who are using P4O-enabled devices to predict the users' usage pattern. The patterns are predicted for 3 days, and hence, there are a total of 576 slots ($48 \times 4 \times 3 = 576$) where the proposed P4O framework predicted the usage pattern. Out of the 576 slots, the P4O framework correctly predicted the usage pattern for the 417 usage pattern slots. Hence, the accuracy can be calculated as follows:

$$\text{Accuracy} = \frac{\text{Total correctly predicted slots}}{\text{Total predicted slots}} \times 100 \tag{4}$$

As per Eq. 3, the accuracy of the proposed P4O framework is 73%. In order to provide further statistics about the performance of the P4O framework, the difference of the predicted versus the actual battery usage pattern data is plotted and is shown in Fig. 11. The difference is plotted as a standard normal distribution. It can be observed from the figure that the predictions in most of the slots are centered on zero (as the mean value). This implies that the predicted value coincides with the actual value of the battery usage at most of the usage pattern slot which further validates the accuracy of the proposed P4O framework.

5.3 Power optimization

In the latest set of experiments, the performance of the P4O framework is tested in real time. The experimental setup is as

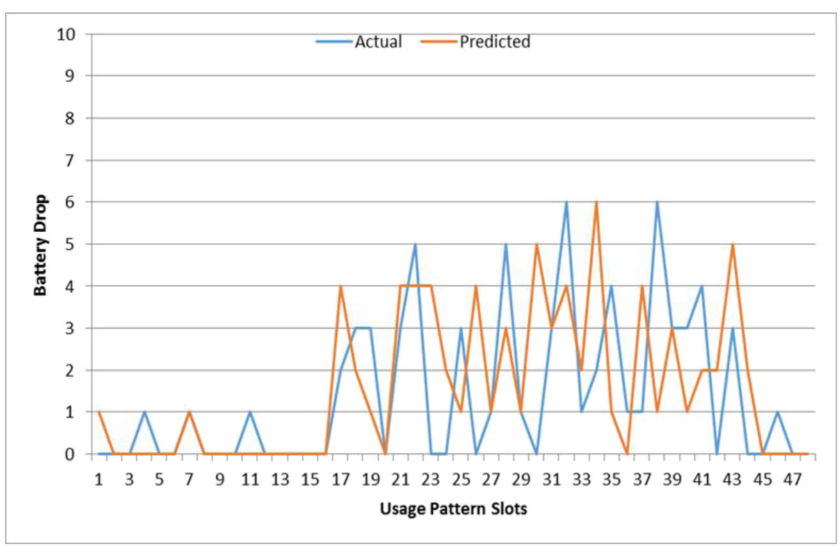


Fig. 10 Power state: actual vs predicted

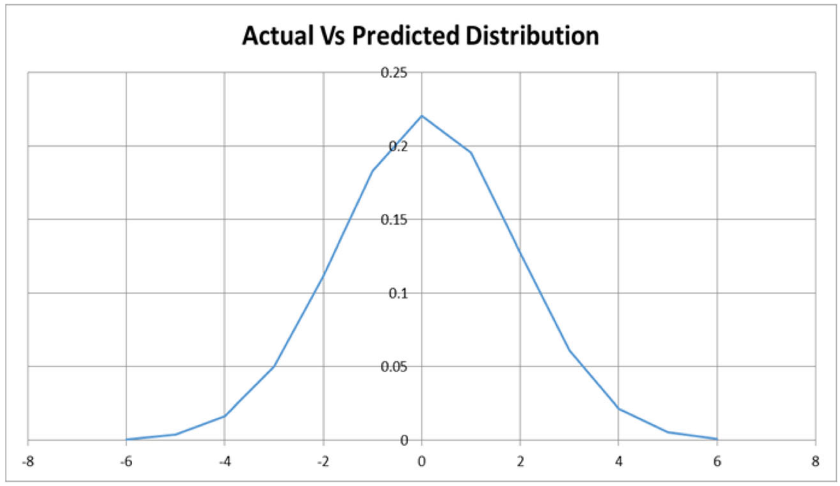


Fig. 11 Power state: actual vs predicted distribution

follows: 10 Samsung Galaxy J2 2016 devices are utilized for experiments. The experiment is divided into three phases. The first phase is the *benchmarking phase*, where these 10 devices, without P4O framework software, are given to 10 users for 1 week. We collected the battery consumption data of all 10 users for each day of the week (Monday to Sunday). The battery consumption data is shown in Table 4. As we can see from Table 4, the battery discharge values are different for every user; at the same time, it differs from day to day for the same user.

The second phase is the *learning phase* where all these 10 devices are enabled with the P4O framework in the learning mode for 2 weeks. As explained in the "Building usage pattern knowledge base" section, the P4O algorithm–learned battery consumption pattern for each of the 48 usage pattern slots during each day of the week (total 48 × 7 = 336 slots). The last phase is the *validation phase* where all these 10 devices trained with the P4O framework are tested for 1 week. Table 5 below shows the battery consumption data in actual devices at the benchmarking phase vs the P4O-optimized devices at the validation phase. As we can see from Table 5 below, the P4O algorithm consistently optimizes the power for each user each day.

It is evident from Table 5 that the P4O-optimized devices have low battery power consumption and hence are capable of saving more battery power in comparison with devices without the P4O framework. A snapshot of the battery level from 3 (out of 10) sample devices from the validation phase is shown in Fig. 12. It can be observed that the proposed P4O framework can learn the user pattern and can effectively use the pattern to optimize the battery power of the device.

The power saved by the P4O framework is further depicted in Figs. 13 and 14. In Fig. 13, the average weekly power savings with the P4O framework for 10 users are shown. On the *X*-axis, the user number is plotted, whereas on the *Y*-axis, the battery drop percentage is plotted. In the figure, bars in blue color show the average weekly battery discharge

Table 4 Daily percentage battery drop for 10 users

User no.	Day of the week						
	Monday (%)	Tuesday (%)	Wednesday (%)	Thursday (%)	Friday (%)	Saturday (%)	Sunday (%)
1	63	65	58	53	70	80	76
2	56	59	65	51	68	76	69
3	45	43	39	54	48	55	57
4	73	71	79	70	68	82	85
5	66	61	68	70	72	56	59
6	60	56	66	51	57	52	50
7	49	43	40	41	41	39	36
8	84	87	89	80	82	75	73
9	56	53	47	53	38	42	40
10	72	67	63	69	75	78	74

Table 5 Daily percentage battery drop vs optimized battery for 10 users

User no.	Day													
	Monday (%)		Tuesday (%)		Wednesday (%)		Thursday (%)		Friday (%)		Saturday (%)		Sunday (%)	
	Bench	Opt	Bench	Opt	Bench	Opt	Bench	Opt	Bench	Opt	Bench	Opt	Bench	Opt
1	63	42	65	46	58	37	53	34	70	47	80	55	76	51
2	56	43	59	46	65	47	51	39	68	54	76	55	69	53
3	45	32	43	34	39	28	54	39	48	36	55	36	57	41
4	73	56	71	55	79	60	70	55	68	51	82	61	85	68
5	66	47	61	46	68	46	70	46	72	58	56	40	59	42
6	60	47	56	45	66	55	51	40	57	45	52	39	50	40
7	49	30	43	27	40	26	41	27	41	26	39	23	36	25
8	84	60	87	70	89	68	80	58	82	64	75	55	73	52
9	56	41	53	38	47	33	53	40	38	29	42	31	40	28
10	72	52	67	46	63	49	69	50	75	56	78	54	74	56

percentages with the original software, whereas bars in red color show the average weekly battery discharge percentages with the P4O software. It is clearly visible from the figure that the P4O framework consistently optimizes power and has less battery consumption for each user. The value of power saved for various users differs from each other. Some of the users use device heavily whereas some users use device moderately (average battery drop minimum 41% for user no. 7 and maximum 81% for user no. 8).

The range of the power saved by the P4O framework for the 10 users for a day is depicted in Fig. 14. As we can see from the graph given in Fig. 14, the P4O framework is able to optimize the power from 20 to 40%. In this graph, the user numbers are plotted on the *X*-axis and their corresponding device power savings (percentages) are plotted on the *Y*-axis. It can be seen from the figure that for user 1, the percentage of power saving is 33.33%, for user 2, the percentage of power saving is 23.08%, and so on.

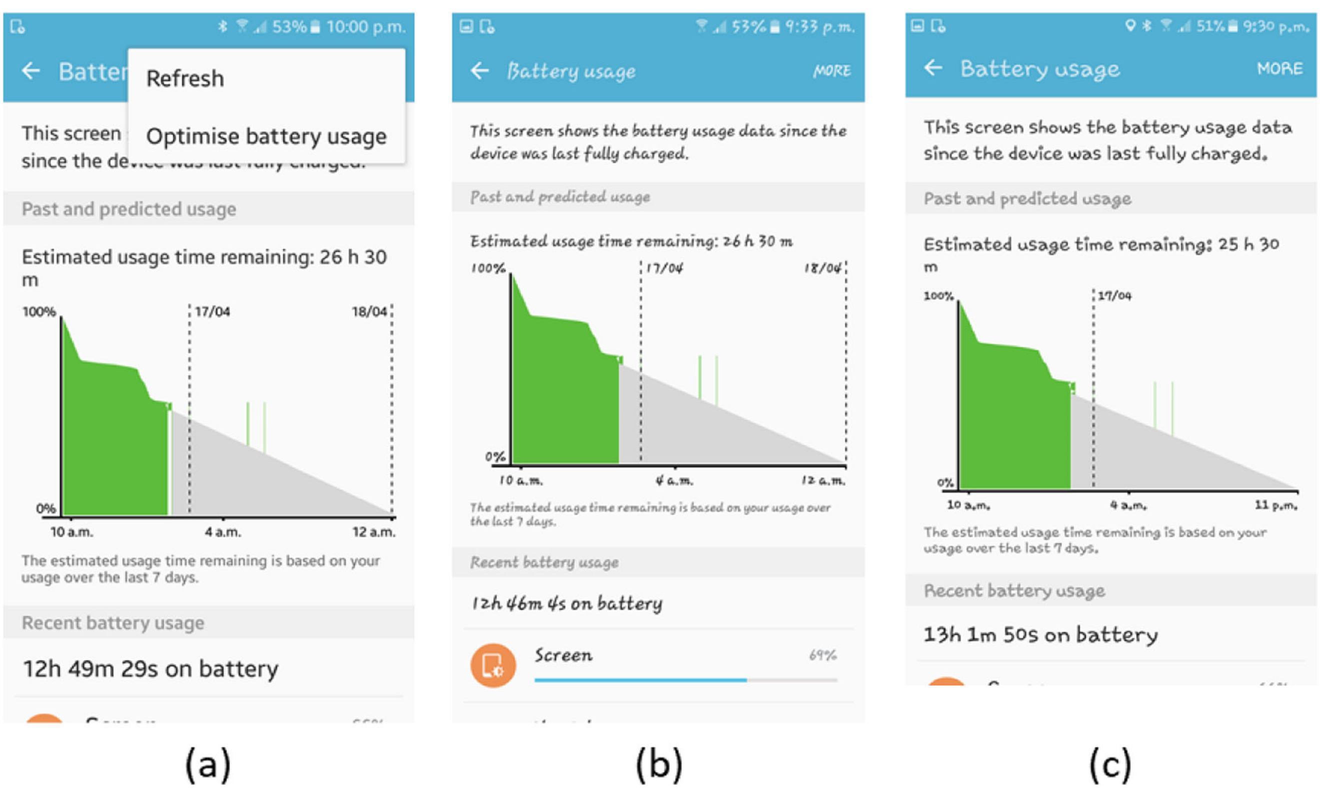


Fig. 12 **a**–**c** Test devices with power savings.

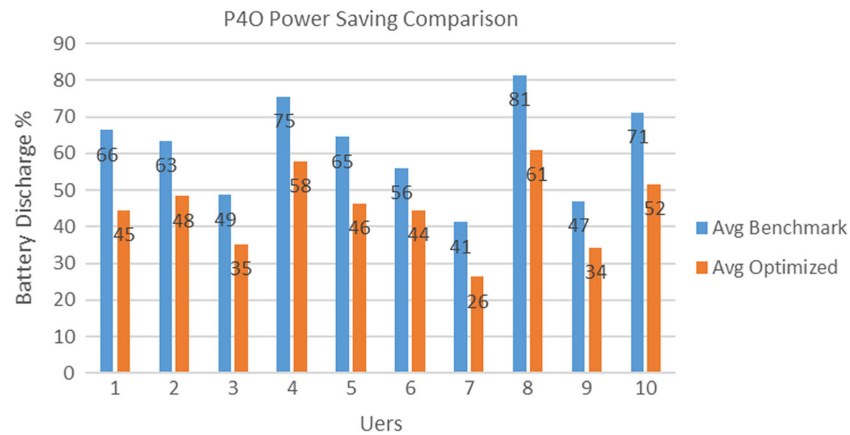


Fig. 13 Average weekly power savings with the P4O framework (10 users)

5.4 Battery lifetime forecast

Battery lifetime forecast is an important aspect of the P4O framework. Based on the recorded battery discharge pattern, the system can accurately predict the time up to when a battery can last without charging. For this experiment, 10 users were given 100% charged devices (Samsung J2) and were asked not to charge their devices until the battery is fully drained. The users noted the number of hours the battery of their devices lasted. For each user, at the start of the experiment, the P4O system predicted the number of hours the battery of that particular user will last based on the knowledge base that the P4O framework has learned. The experimental results are shown in Fig. 15. It can be seen from the figure that for all users, the graphs for the actual number of hours and the predicted hours overlap. From the percentage error graph, it can be seen that the percentage error between the actual battery lifetime hours and the predicted hours ranges from 2 to 26%. Thus, the best-case accuracy of the P4O framework for the battery lifetime prediction is 98%.

6 Conclusion and future work

Power conservation is central to the use of smart phones. The technology in fabrication has kept pace with the times following Moore's law; the battery technology has lagged behind. Therefore, power optimization plays a key role in enhancing the mobile users' satisfaction. To the best of our knowledge, very less or none of the earlier works (SOA) evaluated smart devices in these two directions. A comparative study on some recent state of the art and the present work is given in Table 1. It can be noted that the power optimization in smart phones and battery life forecast is a relatively new area of research. The contributions of this paper are twofold: (1) a novel framework for power optimization in smart phones, referred to as P4O, (2) a new approach for battery lifetime forecast. The proposed framework called P4O (*P*attern, *P*rofiling, *P*rediction, and *P*ower *O*ptimization) is detailed in the "P4O framework" section. This framework has many advantages. It solely runs on the device. Therefore, there is no security or

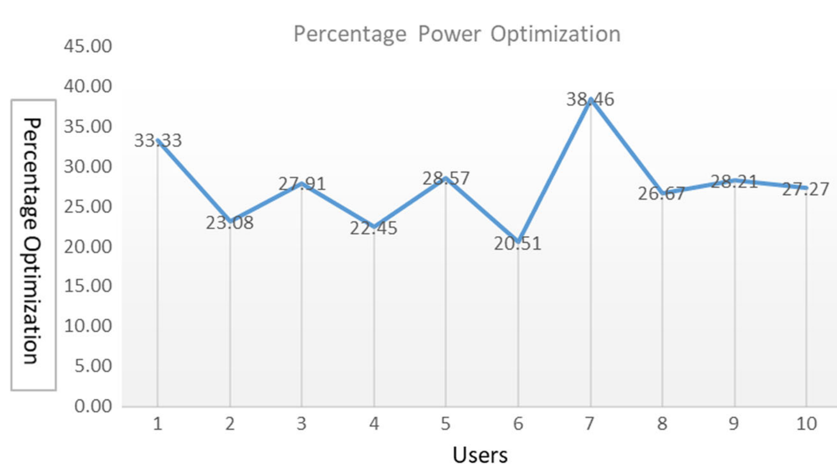
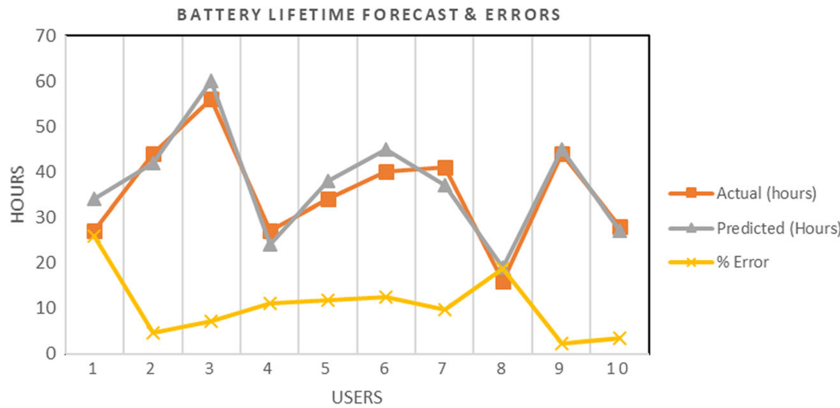


Fig. 14 Percentage of power savings with the P4O framework (10 users)

Fig. 15 Battery lifetime forecast and errors (10 users)



privacy aspects involved in the use of this framework. It directly makes use of the battery discharge rate information for benchmarking user behavior. This makes it independent of which application user uses, as the user tends to install/uninstall applications. This paper also proposed a new approach for battery lifetime forecast (the "Power consumption pattern and power states" section). The proposed method makes an accurate prediction of the battery lifetime.

The proposed framework has been implemented on the Android smart phone. Its efficacy has been validated in experimental test results. We conducted experiments at different stages. In the first set of experiments, users are given four mobile sets without the P4O framework to learn the usage patterns for 2 weeks. In the next set of experiments, we provided P4O-enabled devices to the same set of users to see if the proposed framework is able to predict the learned usage patterns. With this experiment, we established that the proposed P4O framework can predict the battery lifetime with 98% accuracy. In the third set of experiment, we validated the efficacy of the P4O framework as shown in Figs. 11 and 12. The experimental test results validate the proposed P4O framework with the power optimization of 20–40% over the default Linux and Android power saving features in an Android smartphone. More power savings can be achieved by the P4O framework by adding more low-power consuming actions in the power state transitions. This will be investigated in the future.

It may be noted here that this work requires the support of volunteers. We chose volunteers from the staff group of Delhi Technological University, Delhi, India. We carried out the experiments with four volunteers. It is important to note that the less number of probes (volunteers) is due to the nature of experiments. First, it requires a supportive set of volunteers who can judiciously use the device to provide trustworthy learning to the device in order to get reliable results at prediction. Second, the experiments need to be carried out for a longer period of time (at least 2 weeks). It requires large-scale effort to engage a huge number of volunteers for such a longer period of time while the scope of this manuscript is to introduce the proposed P4O framework and prove its efficacy with experimental results. Thus, to provide experiments on a large volume of users is in the scope of the future work.

References

1. Gartner Report, Egham, UK, February 15, 2017. Available at: http://www.gartner.com/newsroom/id/3609817. Accessed 27 March 2019
2. Deloitte 2015 Global mobile consumer survey. Available at: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/technology-media-telecommunications/us-tmt-global-mobile-executive-summary-2015.pdf. Accessed 27 March 2019
3. Samsung Galaxy Note 7 with inbuilt Iris scanner. https://news.samsung.com/global/everything-you-need-to-know-about-the-galaxy-note7s-iris-scanner. Accessed 27 March 2019
4. Samsung Galaxy S7 Edge specifications. https://www.techradar.com/reviews/phones/mobile-phones/samsung-galaxy-s7-edge-1315189/review. Accessed 27 March 2019
5. IObit Applock face recognition unlock Android application. http://www.iobit.com/en/applock.php. Accessed 27 March 2019
6. Phone user behaviour report for unlocking phones. http://www.phonearena.com/news/Did-you-know-people-unlock-their-phones-an-average-100-times-a-day_id74189. Accessed 27 March 2019
7. Rahmati A, Qian A, Zhong L (2007) Understanding human-battery interaction on mobile phones. In: Proceedings of the 9th international conference on human computer interaction with mobile devices and services. ACM, Singapore
8. Balani R (2007) Energy consumption analysis for Bluetooth, Wi-Fi and cellular networks. White Paper, Tech Republic
9. Rice A, Hay S (2010) Decomposing power measurements for mobile devices. In: IEEE International Conference on Pervasive Computing and Communications. Manheim, Germany 29 March – 2 April 2010
10. Carroll A (2010) An analysis of power consumption in a smartphone. In: Proc. USENIX Annual Technical Conference. Boston, MA, USA, 23–25 June 2010
11. Perrucci GP, Fitzek FHP, Widmer J (2011) Survey on energy consumption entities on the smartphone platform. In: 73rd IEEE Vehicular Technology Conference (VTC). Yokohoma, Japan, 1–5 May 2011

12. Paul K, Kundu TK (2010) Android on mobile devices: an energy perspective. In: 10th IEEE International Conference on Computer and Information Technology (CIT). Bradford, UK, 29 June – 1 July 2010
13. Lyons K, Hightower J, Huang EM (eds) (2011) Understanding human-smartphone concerns: a study of battery life. Pervasive 2011, LNCS 6696. Springer-Verlag, Berlin Heidelberg, pp 19–33
14. Demumieux R, Losquin P (2005) Gather customer's real usage on mobile phones. In: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services. ACM, pp 267–270
15. Kang JM, Seo S, Hong J (2011) Usage pattern analysis of smartphones. In: 13th Asia-Pacific Network Operations and Management Symposium. pp 1–8
16. Oliver E (2010) The challenges in large-scale smartphone user studies. In: Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement. ACM, pp 1–5
17. Lee J, Joe H, Kim H (2012) Smart phone power model generation using use pattern analysis. In: IEEE International Conference on Consumer Electronics. pp 412–413
18. Falaki H, Lymberopoulos D, Mahajan R, Govindan R, Kandula S, Estrin D (2010) Diversity in smartphone usage. In: Proc. ACM MOBISYS
19. Verkasalo H (2010) Analysis of smartphone user behavior. In: Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), Ninth International Conference on IEEE, 2010, pp 258–263
20. Oulasvirta A, Rattenbury T, Ma L, Raita E (2012) Habits make smartphone use more pervasive. In Pers Ubiquit Comput 16:105–114
21. Adrakatti AF, Mulla KR (2017) A realistic approach to information services on mobile apps. J Access Serv 14(1)
22. Min AW, Wang R, Tsai J, Ergin MA, Tai TYC (2012) Improving energy efficiency for mobile platforms by exploiting low-power sleep states. CF'12, may 15–17, Cagliari, Italy
23. Al-Turjman F (2017) Cognitive-node architecture and a deployment strategy for the future sensor networks. Mobile Networks and Applications, Springer, ISBN 9781138102293 - CAT# K34952. https://doi.org/10.1007/s11036-017-0891-0
24. Singh GT, Al-Turjman FM (2016) Learning data delivery paths in QoI-aware information-centric sensor networks. IEEE Internet Things J 3(4):572–580
25. Kang J-M, Seo S-s, Hong JW-K (2011) Personalized battery lifetime prediction for mobile devices based on usage patterns. J Comput Sci Eng 5(4):338–345
26. Krintz C, Wen Y, Wolski R (2004) Application-level prediction of battery dissipation. In: Proceedings of the International Symposium on Lower Power Electronics and Design. Newport Beach, CA, pp 224–229
27. Zhang L, Tiwana B, Dick RP, Qian Z, Mao ZM, Wang Z, Yang L (2010) Accurate online power estimation and automatic battery behaviour based power model generation for smartphones. In: Proceedings of the 8th IEEE/ACM International Conference on Hardware/Software-Co-Design and System Synthesis. Scottsdale, AZ, pp 105–114
28. Ferdous R, Osmani V, Mayora O (2015) Smartphone app usage as a predictor of perceived stress levels at workplace. In: 9th IEEE International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health). Istanbul, Turkey
29. Al-Turjman F, Betin-Can A, Ever E, Alturjman S (2016) Ubiquitous cloud-based monitoring via a mobile app in smartphones: an overview. In: IEEE International Conference on Smart Cloud (SmartCloud). New York, NY, USA
30. Elgedawy I, Al-Turjman F (2016) IdProF: identity provisioning framework for smart environments. In: IEEE Conference HONET-ICT. Nicosia, Cyprus
31. Li H, Lu X, Liu X, Xie T, Bian K, Lin FX, Feng, F (2015) Characterizing smartphone usage patterns from millions of Android users. In: Proceedings of the 2015 ACM Conference on Internet Measurement Conference, pp 459–472
32. Lu EHC, Lin YW, Ciou JB (2014) Mining mobile application sequential patterns for usage prediction. In: 2014 IEEE International Conference on Granular Computing (GrC). Noboribetsu, Japan, pp 185–190
33. Zipf GK (1965) Human behaviour and the principle of least effort: an introduction to human ecology. Hafner Publishing Co., New York
34. Google Android developers information. Available at: https://developer.android.com/index.html. Accessed 27 March 2019
35. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82(1):35–45
36. Monsoon Power Monitor by Monsoon Solutions Inc. Available at: https://www.msoon.com/LabEquipment/PowerMonitor/. Accessed 27 March 2019
37. Samsung Galaxy J2 (2016) Available at: http://www.samsung.com/in/smartphones/galaxy-j2-2016-j210f/SM-J210FZDDINS/. Accessed 27 March 2019

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.