IJournals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 4 Issue 4 April, 2016

UIPE: Increasing Power Consumption Efficiency of Smart Devices Using Usage Pattern

Author: Md. Arif Ibne Ali[1]; Rayhan Hossain[2]; Rayhanul Islam[3]; Rayhanur Rahman[4]
Institute of Information Technology, University of Dhaka, Bangladesh

arifibneali@gmail.com[1];hossain.rayhan@outlook.com[2]; bit0203@iit.du.ac.bd[3],rayhan@du.ac.bd[4]

ABSTRACT

A lot of dependencies have increased in the growing phenomenon of smart devices such as Smart phone, Tabs, etc. Those devices play a significant role in the development of world through mobile computing facilities like e-commerce, socialmedia, etc. To gain maximal service from smart devices, battery limitations is the great concern. This paper describes an approach named UIPE (User can Increase Power Efficiency) to prolongbattery life. At first, UIPE collects data from user activity and then usage pattern is generated using our proposed pattern generation technique based on those collected data. After that, smart power saving profile is produced based on the generatedusage pattern for that particular user in profile module. Finally, profile activator module activates power saving profile based on usage pattern. Besides those, UIPE also provides severalbest practices for the user to expand valuable battery life. The battery level gain using UIPE is compared with two well-known power saving android applications namely DU Battery Meter andJuiceDefender. Experiment shows that UIPE increased nearly 22% of battery life compared to above mentioned applications.

Keywords- Power Efficiency, User Behavior, Battery Saving,Smart Mobile Devices

1. INTRODUCTION

Mobile devices have immense popularity among various kinds of peoples by connecting to Internet and providing valuable information through multiple applications [1]. It is found that 0.4 million android devices are being activated in the daily basis [2]. Over the past few years, exponential development has been taken place in the Smart device capacities by integrating high processing power, rich sensors, better screen determination and fast communication over Wi-Fi, 3G and LTE. People's web surfing habits have changed because of smart device multitasking ability and its high computing capability. It introduces huge advancement of technology and people considered it as most valuable demographic information. E-commerce, social media, location based marketing, gaming, corporate communications, global news, e-health, entertainment has gone smart device.

Although high computing power and capabilities are available in new technology and trends, those devices still sicken from battery limitations and its management. Power usage of smart devices mainly depends on user behavior and interactions. Intelligent optimization of software and hardware based on the battery status is the main key to save the energy of the smart devices. The aim of our research is to increase power consumption efficiency of smart devices by analyzing usagebehavior and user interactions. This research introduces a proficient approach to increase power consumption efficiency.

There are lots of approaches to solve the power problem in smart devices. One of the approaches to solve the power problem by analyzing program analysis [3].It used a combination of program analysis and energy modeling. This research is mainly developer centric to help them to understand apps energy consumption behavior. There are lots of surveys, user centric research has been conducted in this domain [2], [4], [5]. Proposed method of this paper is a new technique to save battery life of the smart devices.

This intelligent approach, UIPE contains four segments that get data from user (monitoring phase), generate usage pattern using usage pattern generation algorithm [6], smart power saving profiles are recommended based on usage pattern and finally profile will be activated into the device. This power saving profile will change over time if usage pattern willchange with certain period of time. User interaction with the device is main concern of this research. Because power consumption has principally depends on end user activity. UIPE has great control on various feature and functionality of android devices by which it can prolong the life of the battery. It is assumed that user will do a job in particular day based on what he/she did in the same day of the previous week. We have considered three sample devices to conduct our study about power consumption and increasing its efficiency. Three well known popular smart phones have been used to find the efficiency of the power saving applications [7].

We have collected necessary data from these three devices for continuous four weeks. In first week we have collected power-related information from those three devices without installing any power saving applications. In the second, third and fourth week we have collected the information afterinstalling three power saving applications DU Battery Meter [8], Juice Defender [9], [10] and our UIPE respectively. Then in the fifth week we started to generate UIPE user profile based upon the collected data of fourth week. Here we have used our profile generating algorithm for the advancement of thebattery life. Some user guidelines were also provided for UIPE user to make the device less power hungry. After applying ourgenerated user profile we measured and compared the power efficiency results gathered from three devices. Finally we havefound UIPE better than other two applications. To increase the accuracy of our study we have chosen three users from three different professions. One of them was a university lecturer, one was a software engineer and the last person was a student. Result shows that the proposed approach UIPE outperformsthe existing two well-known applications named DU BatteryMeter and Juice Defender. UIPE increased noticeable amount of battery life compared to above mentioned applications.

The rest of the paper is organized as follows: Related work is discussed in section 02, section 03 presents our overall methodology, 04 describes results and evaluation of UIPE. Finally we conclude our research in section 05.

2. RELATED WORK

There have been several studies on power consumption, energy management and their efficient usage. Significant amount of research have been conducted to understand how smart devices use its power according to the user behavior. So, the increasing power consumption efficiency using usage pattern is definitely a flourishing area of research. In this section, we review the current state of the art connected to our research.

Soumya et al. [6], [11] stated that power consumption hugely relies upon end user interaction. They have proposed a client-server architecture for introducing personalized power saving profile by analyzing the individual usage pattern. The whole experiments were performed using an android app “Power Monitor”. To generate usage pattern, pattern generation algorithm was applied in the server to create profiles and those are sent back to device. They have shown that power saving profile highly depends on the usage patterns and security and management issues of client-server architecture. Results showed that it increases battery life by 90 percent. Though it worked well, it lacked the user satisfaction and also inherited the flaws of client-server architecture.

Joon-Myung et al.[12] have defined five possible states of a smart phone based on its basic operation. They have presented a case study in order to show: how to apply usage pattern information to smart phone power management, device and network management. They have provided the statistics of 20 smart phone user activities based on five basic operations. They have developed a Cumulative Distribution Function (CDF) for three main operational states (Calls, Wi-Fi and 3G) and finally introduced a case study for the usage pattern based battery lifetime prediction. The main drawbacks of this paper were it only considered network management avoiding impact of others.

Soumyaet. al.[13] depicted self-adaptive strategies in implementation level and how to dissipate power from android devices. They have proposed three power saving profiles with
different self-adaptive features through self-adaptive application framework in mobile application development. This framework has gathered hardware and context information that considered in this research. It categorized all devices into three profiles but sometimes it looked difficult to categorize a device into those mentioned profiles.

Android power management based survey was conducted on several power saving apps like juiceDefender, SetCPU [2]. Result found that profiles were statically defined and controlling connectivity and brightness were not very intelligent.

All the above experiments discuss about the power efficiency of mobile devices. But all work cannot meet the all requirements. This manuscript incorporates all work to increase the power efficiency.

3. POWER SAVING PROFILE GENERATION APPROACH ACCORDING TO USAGE PATTERN

3.1 Collecting User Information

Power consumption of smart devices predominantly depends on how resources of the devices are being used. Applications of the device are the fundamental processes that run upon android operating system. Applications also consume noteworthy amount of battery as well hardware resources. We can extract how much power is consumed by hardware andapplications observing user behavior. To get usage information we have developed a monitoring module by which we can collect all the necessary data about user interaction andhardware resources. It records how the smart device is being employed in daily basis. All the information collected from monitoring module are stored to the local memory. Primarily monitoring engine collects data for a week from device to generate usage behavior of the user.

Next week monitoring module again collects data from device and compares with previous weeks data. If any remarkable change is detected, the usage information will be also updated that stored in the local database. Android database SQLite has been used to store the monitoring information.Monitoring information is used by pattern generation algorithm to suggest appropriate profile for the device to save power drain. Power saving profiles is applied to android devices to increase the efficiency of the battery life. After getting final power saving profile user will get notification. User can continue with the giving profile or can make it personalized as he needed. Like [6] our module is also having an activation time, a termination time and a list of settings. Changes of usage pattern have strong impact on generating power saving profile. To apply appropriate profile in appropriate case depends on user behavior and choice.

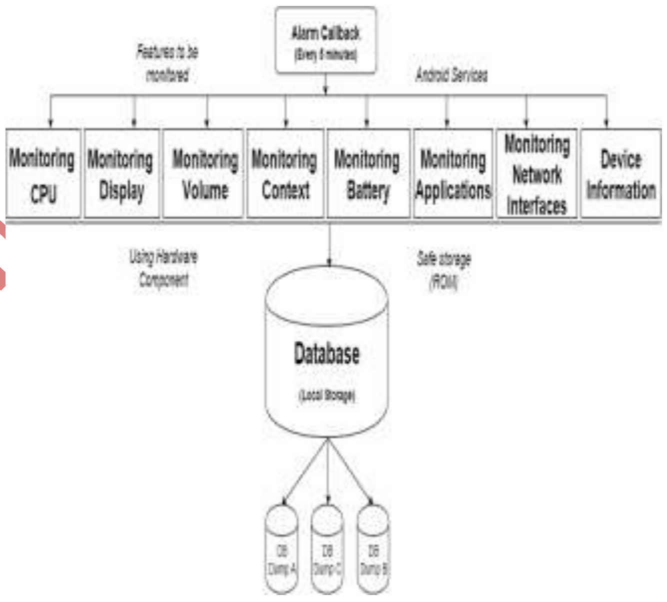

Fig. 1: UIPE Monitoring

3.2 Monitoring Module

Efficient monitoring of usage behavior is important to support smart power management. To derive usage pattern acomplete monitoring of the device is very essential. Monitoring module thoroughly collects data adhering to user behaviorof the device. Figure 01 imprints the architecture of the monitoring module which is constructed using background service. Background service of android has collecting data in every five minutes for seven continuous days (one week). Service continuously makes some valuable information. This information contains different types of user data. Data consists of hardware resource use information, time serous values of several features of the device. Fetched data are stored locally in the device storage. Android alarm manager callback provides access to the system alarm services. Alarm callback allows system to schedule android application to be run at some point in the future. It has been used here, to suspend the service between two callbacks [6]. The monitoring module collects device location via mobile network. Network device, GPS has also can obtain current location of the device. But our module collects location via mobile network. It consumes less power. On the other hand GPS and other network device is very power hungry features. In our research device location is very important because locations are used to demark the power saving profile with a point in time.

TABLE I: Monitoring Module of UIPE

Module Name	Feature of the Device
CPU Monitoring	• CPU load
Monitoring Applications	•Number of running applications
Monitoring Battery Behavior	• Battery level(%) •Battery status (charging/discharging)
Display Monitoring	• Brightness • Screen timeout • Total interaction time
Monitoring Location and Context	• Location of the device • Current date • Time
Audio Information Monitoring	• Media volume level • Alarm volume level • Phone volume level
Network Features Monitoring	• GPS status • Wi-Fi • Mobile data
Device Details	• Operating system version • Device model • Manufacturer

Smart device features that should be considered are listed into Table 01.

3.2.1 Monitoring CPU:

To find out how much battery is being consumed by CPU, system has to monitor it properly. Power drain of smart device highly depends on the CPU usage. The monitoring module calculates total load of CPU continuously. All the running application will be listed by the profiler. User can kill any application that seems less important to him/her. It has great impact on the battery life because higher CPU load can cause higher battery expenditure [6].

3.2.2 Running Application Monitor:

Amount of running applications can be listed by the monitoring module. Duration of running applications and engagement time of the user are the two main reasons of draining battery. For example, gaming, social media apps need higher user engagement and therefore CPU and display consumes considerable amount of energy. But it is proved that high interaction time doesn't mean high battery consumption [12].

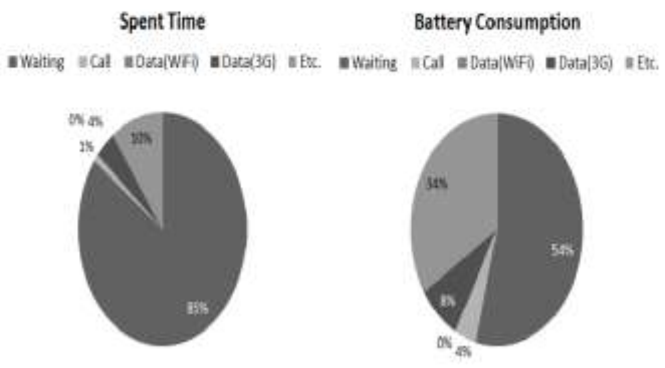

Fig. 2: Spent Time vs Battery Consumption

3.2.3 Monitoring Battery Behavior:

Battery health and its efficient usage is the main concern of this research. Battery level and battery status [12] will be monitored in this phase. In every DP time, there are start time battery level, average battery level and ending time battery level. This module can collect all the battery level information of the device. It can also collect charging and discharging pattern, called battery status of the smart device.

3.2.4 Record Display Usage:

Screen brightness and screen timeout information are also recorded by the monitoring module. These two consumes significant amount of power. Interaction with the device is the only way to control it. This phase of the monitoring module can record the total interaction time of the device.

3.2.5 Location and Context Monitoring:

This phase has identified the location of the device, system date and time. To determine a unique pattern for a distinct position, location and context monitoring is very important.

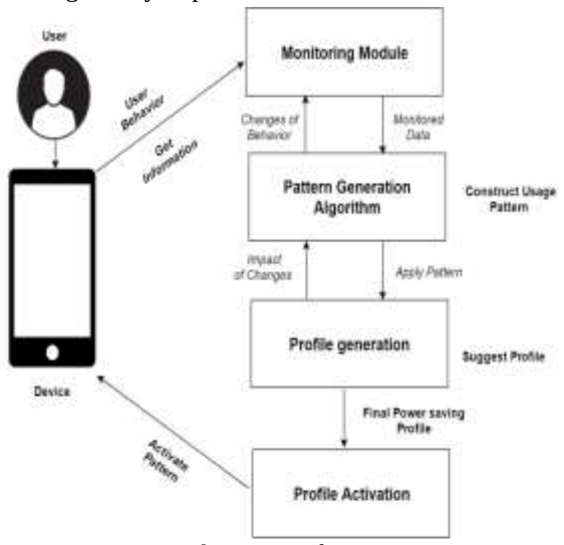

Fig. 3: Architecture of UIPE

3.2.6 Monitoring Audio Information:

Three types of audio information (call volume, media volume and alarm volume) are observed by the monitoring module. These different types of volume level can affect battery life immensely.

3.2.7 Monitoring Network Features:

Maximum usage of networking feature causes the maximum consumption of battery. GPS, 3G/2G network, Wi-Fi has been monitored in this step.

3.2.8 Device and Operating System Information:

This is a secondary feature of the monitoring module. It collects all the device information like OS version, manufacturer, model number, etc.

3.3 Usage Pattern Generation Approach

Monitoring module stores its monitored data into a local database. After one week, the stored data in the local database is sent to be stored into a dump database. A number of dump databases may exist in the system for saving the weekly data. Then the following processing will be performed for generating usage pattern.

Necessary inputs are taken from the local database to begin the usage pattern algorithm. Every users usage patterns is distinguished based on three major parameters [13], [6].

- A specific a day of the week (d)
- Time interval of a day (t)
- Location of the user (s)

All these are collected using the context monitoring features. Based on these three parameters, a pattern can be generated for a particular device using the following algorithm.

1) Algorithm: Processing of this algorithm will run inside the device. Every day of week will repeat the following:

1. Count every independent location of the user,(s1,s2,…,sn)
2. Demarcate the time interval (t) linked with every independentlocation. Same location can be repeated withvarious time intervals in the different parts of the day
3. Make a pair of Time Interval and Location. Repeat 4-13which will develop the respective usage pattern of theuser
4. Identify the number of running applications
5. Determine the overall CPU load
6. Determine battery level in different span of time (startbattery level, average level, end battery level)
7. Identify battery status (charging/ discharging)
8. Identify volume level (phone volume, media volume,alarm volume)
9. Calculate the overall network usage (3G/2G network,Wi-Fi)
10. Enumerate the usages of Blue-tooth and GPS
11. Determine the total interaction time
12. Record the brightness level
13. Record screen timeout value

This algorithm is activated after getting information from the monitoring module and it runs within the device. The usage patterns with similar time continuation and same location over various days are amalgamated together. The first objective of this pattern generation algorithm is to estimate the power consumption for informing the user. There are also some other factors like high user engagement time (device interaction time), higher internet usage, battery charging pattern, severe acute CPU operations that have effect on the pattern creation. Finally, from the gathered information, the usage pattern generation algorithm suggests a smart power saving profiler for the device user.

3.4 Power Saving Profile Generation Algorithm

Power saving profiles is the set of predefined settings that govern the internal mechanism of a smart device. Based on the user information, smart power saving profile triggers itself and employs its capability to the device. A smart profiler mainly regulates CPU load, user interaction, volume, vibration, etc. of the device [14]. Data transfer and communication interfaces like Bluetooth, Wi-Fi, network and location interfaces like GPS are also regulated by the smart profiler. The usage pattern analyzer generates the usage pattern of a user of the mother device, by getting information from the monitoring module. Finally, the smart profiles are activated to reduce the power consumption after achieving the user behavior of the smart devices. Similar to [13], the profiler checks the following parameters before generating the user profile.

- Battery status: It denotes whereas mobile is in charging or discharging mood.
- Battery level: It counts the percentage of the battery energy level.
- Charging behavior: It estimates the next charging opportunity or behavior of the device in the selected day considering the previous charging behavior.

After examining the above three parameters, the following algorithm creates a gross smart power saving profile. The algorithm is executed in the mother device where the systemhas been installed and run (including pattern generation algorithm). The profile generation algorithm is written below:

1. Firstly, the monitoring module categorizes computationally heavy weight apps, characterized by high CPU load and operating frequency. If the device battery level is below 50% and the CPU usage is above 50%, thenthe users are given a choice to kill less necessary applications.
2. If the user has spent more than 60% of the pattern time with the device, then the pattern must be registered as high device interaction time through pattern generation algorithm. High device interactions cause high energyconsumption. Based on the device interaction time, the brightness level and screen time out values will be reduced. Table II illustrates necessary actions about Screen Timeout (ST) and Brightness (B) while the user interaction time is high (Assume: more than 1.5 hour).

TABLE II: Necessary Steps: Screen Time out (ST) and Brightness (B)

Battery Level **(%)**	Battery Status: Charging **Actions: Decrease By(%)**	Battery Status: Discharging **Actions: Decrease By(%)**
Battery Level: <20%	ST: (-80%) B: (-70%)	ST: (-90%) B: (-80%)
Battery Level: >20% and <50%	ST: (-60%) B: (-50%)	ST: (-70%) B: (-60%)
Battery Level: >=50%	ST: No Change B: No Change	ST: (-30%) B: (-50%)

3. Now, if the user uses less than 20% of the total pattern time, we can consider it as deficient interaction time. If the pattern records this kind of less interaction with the device, the Wi-Fi, 3G data connection should be switched off as the user is not using it. The user can turn it on again later while he wants to interact.
4. If there are 2.0 hours of no interaction with the device, it is considered as idle time (no additional activity) of the smart device. It can happen during the night or any sleeping time. So, in the no interaction time, the connections without mobile network will be switched off which will save a noticeable amount of energy.
5. Bluetooth is another power hungry feature of smart device. If monitoring module has found that it has not been used for a noticeable amount time, it is turned off.
6. Provide the choices to kill the following when battery drains in striking rate.
   - Audible touch
   - Notification tone
   - Live wallpaper etc.

7. Media volume (YouTube), call volume (phone), device volume (tone, alarm) - these three types of

smart device volumes are considered. Considering Table II, profiler will take the following actions to decrease the Volume level (VL) automatically, as shown in Table III.

TABLE III: Necessary Steps: Volume

Battery Status: Charging Actions: Decrease By(%)	Battery Status: Discharging Actions: Decrease By(%)
VL: (-60%)	VL: (-70%)
VL: (-40%)	VL: (-50%)
VL: (-10%)	VL: (-30%)

8. Battery level below 15% is considered as critically low. If it happens, the profiler will take the following actions:

- Dim the screen brightness. Set it to 15%
- Set Time out value to be 05 second
- Turn off all internet connection
- Turn off vibration
- Turn on favorite contact list (This list contains some special numbers who can only be called)
- Assure less use of CPU
- Dump unnecessary widgets and live wallpaper
- Turn off sensors and GPS

9. Auto-sync will be activated during remaining battery level of 70%-90%. This feature can be configured by the user or the application (Drop box, Google drive, pocket etc.).
10. If GPS is used for long time (30% of the total pattern time), profiler will ask the user to turn it off as GPS cannot be turned off automatically [6].
Monitoring module gets necessary information by analyzing the user behavior. Because the user behavior evolves over time and it has strong impact on usage pattern generation. Every pattern is activated with a suitable profile. Profile is activated at the beginning of the pattern and remains active till the pattern is alive. If the usage pattern has any changes, another profile will be created and re-activated. These phases will be continued in the mother device to increase the efficiency of the smart device energy.

4. IMPLEMENTATION AND RESULT ANALYSIS

A. *Environment Setup*
This section discusses the tools and equipment's used to develop the methodology and experimental procedures followed for the evaluation of this research. Methodology was developed in Android and deployed in a smart device. In order to develop the methodology, following tools are used:

- Android Studio

Developed methodologies were performed on the following device configuration:

Device Name	**Nexus 7**	**Motorola Moto 3g**	**Symphony W128**
Processor	Quad-core 1.5 GHz Krait	Quad-core 1.4 GHz Cortex A53	Quad-core 1.3 GHz Cortex A7
GPU	Adreno 320	Adreno 306	Mali 400
Device Display	1200 x 1920 px (323 ppi)	720 x 1280 px (294 ppi)	540 x 960 px (294 ppi)
RAM	2GB	2GB	1GB
Operating System	5.0.1	5.1.1	4.2.2
Battery Capacity	4325 mAh Li-Ion	3950 mAh Li-Ion	1750 mAh Li-Ion

B. *Result and Evolution*
After finishing the development the app was installed into three devices (Nexus 7, Motorola Moto g 3rd generation and Symphony W128). Then the data was collected using the monitoring module and respective power saving profile was generated using custom profile generation algorithm. The profile settings were applied to the devices and power efficiency gain was measured in terms of battery level. The gain using UIPE was compared with the gain using another two popular power saving applications available in the Google Play named Juice Defender[13] and DU Battery Saver[14]. Most of the cases performance using UIPE was better than other two applications. Here one performance statistics is given for a specific time slot of a specific day of the week. To measure the performance of our application we compared

the increased battery life using our application with two other applications result named Juice Defender and DU Battery Meter. We have chosen three android phone users with three different phones from different professions. To measure the average efficiency increment we installed the applications to the user's phones and observed result for four weeks. First week we ran the phone without any application, then Juice Defender, DU Battery Meter and our Application respectively for three weeks. Then we measure the average increased battery lifetime for those three applications. For calculating increased battery lifetime using our application we used the following equation (1).

$$\frac{\sum_{i=1}^{day}\left(\frac{\sum_{j=1}^{pattern(timeslot)} EfficiencyIncreased_j}{pattern}\right)_i}{day} \quad (1)$$

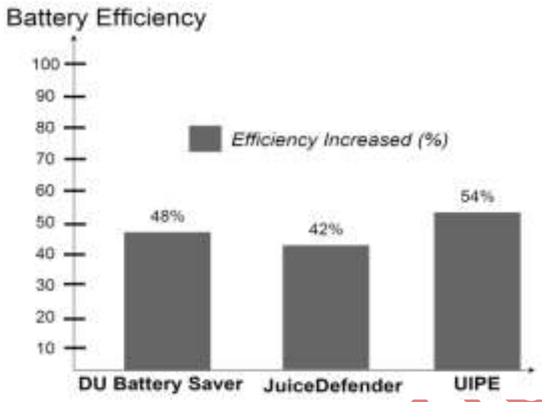

Fig. 4: Nexus7 Result Analysis

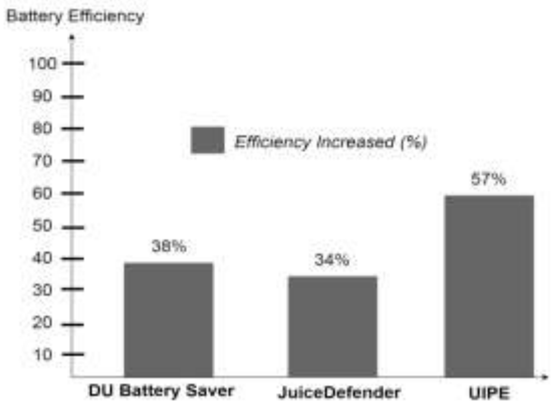

Fig. 5: Motorola Moto g 3rd Gen. Result Analysis

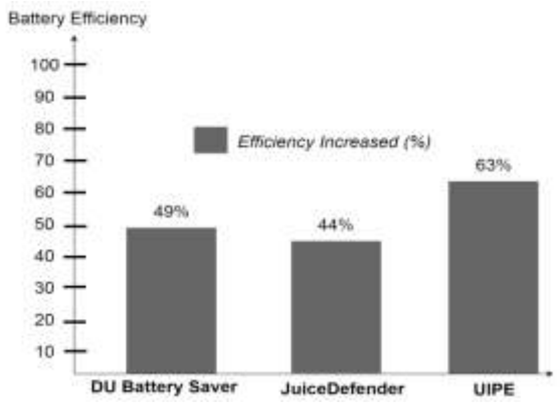

Fig. 6: Symphony W128 Result Analysis

Here first we calculated the total time slots for every users and increased battery lifetime for each slot. Then we calculated the average improvement for a day. We followed the same approach for seven days and after that calculated the average improvement for those seven days by dividing the summation by seven.

5. CONCLUSION

UIPE is a stand-alone application running inside the smart device. There are three major parts of this application. In the first phase, device is monitored by monitoring module. This module collects the information of user activities with the device. Some addition information like location, day, time of the device are also collected. After collecting those, user behavior pattern generation algorithm generates a usage pattern of the device user. Based on this pattern, UIPE will suggest a power saving profiler for the device to prolong its battery life. Moreover, UIPE provides a set of instruction to user to save the device power. Complete comparisons of UIPE with two other very popular power saving applications are provided. The use UIPE of is beneficial because it is able to increase at least 22% of the battery life than two very popularpower saving applications.

REFERENCES

[1]. F. Vinson, "Top 10 Reasons Why Mobile Technology is More Important than Ever," http://http://www.localorganicrankings.com/top-10-reasons-why-mobile-technology-is-more-important-than-ever/, Jan 28 2013, accessed: 2015-09-26.

[2]. S. K. Datta, C. Bonnet, and N. Nikaein, "Android Power Management: Current and

Future Trends," in Proceedings of the First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT). IEEE, 2012, pp. 48–53.

[3]. S. Hao, D. Li, W. G. Halfond, and R. Govindan, "Estimating mobile application energy consumption using program analysis," in 35th International Conference on Software Engineering (ICSE). IEEE, 2013, pp. 92–101.

[4]. J.-M. Kang, C.-K. Park, S.-S. Seo, M.-J. Choi, and J. W.-K. Hong, "User-centric Prediction for Battery Lifetime of Mobile Devices," in In Challenges for Next Generation Network Operations and Service Management. Springer, 2008, pp. 531–534.

[5]. E. Oliver, "Diversity in smartphone energy consumption," in Proceedings of the ACM workshop on Wireless of the students, by the students, for the students. ACM, 2010, pp. 25–28.

[6]. S. K. Datta, C. Bonnet, and N. Nikaein, "Personalized Power Saving Profiles Generation Analyzing Smart Device Usage Patterns," in Proceedings of the 7th International Conference on Wireless and Mobile Networking Conference (WMNC). IEEE, 2014, pp. 1–8.

[7]. J. CORPUZ, "10 Best Mobile Battery Apps," http://www.tomsguide. com/us/pictures-story/711-best-battery-saving-apps.html/, Oct 16 2015, accessed: 2015-10-13.

[8]. D. A. STUDIO, "DU Battery SaverPhone Charger," https://play.google. com/store/apps/details?id=com.dianxinos.dxbs &hl=en/, Sep 26 2015, accessed: 2015-07-02.

[9]. L. Calle della Regina, "JuiceDefender - battery saver," https://play. google.com/store/apps/details?id=com.latedroi d.juicedefender&hl=en/, Jan 05 2012, accessed: 2015-07-02.

[10]. C. Moor, "Extend your battery life with Juice Defender for Android," http://www.talkandroid.com/ 9599-extend-your-battery-life-with-juice-defender-for-android/, Aug 08 2010, accessed: 2015-08-19.

[11]. S. K. Datta, C. Bonnet, and N. Nikaein, "Power Monitor v2: Novel Power Saving Android Application," in 17th International Symposium on Consumer Electronics (ISCE). IEEE, 2013, pp. 253–254.

[12]. J.-M. Kang, S.-s. Seo, and J. W.-K. Hong, "Usage Pattern Analysis of Smartphones," in Proceedings of the 13th Asia-Pacific Symposium of Network Operations and Management(APNOMS). IEEE, 2011, pp. 1–8.

[13]. S. K. Datta, C. Bonnet, and N. Nikaein, "Self-adaptive Battery and Context Aware Mobile Application Development," in Proceedings of th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2014, pp. 761–766.

[14]. NRDC, "How to Reduce Your Energy Consumption," http://www.nrdc. org/air/energy/genergy.asp/, May 28 2015, accessed: 2016-02-16.