Modeling User Activity for Conserving Power on Smartphones

Kumar Vishal, Romil Bansal, Anoop M. Namboodiri and C. V. Jawahar
International Institute of Information and Technology, Hyderabad, India

***Abstract*—Mobile consumer-electronics devices, especially phones, are powered from batteries that are limited in size and therefore capacity. This implies that managing the battery power is paramount in such devices. As long as the battery technology continues its slow pace of improvement, the only viable approach is to reduce the amount of energy required to provide specific services. Two of the most power consuming services on a smartphone are network and wireless data. Though internet connectivity is important, the nature of data transfer does not require uninterrupted service. We take advantage of this fact to cut off power to these modules when it is not required. The challenge is to predict, when the users require these services and when they do not. Existing solutions usually schedule the services based on fixed hand-coded rules that do not work for all users or usage scenarios. In this paper, we propose an activity-based model that uses machine learning approaches to intelligently schedule Wi-Fi according to a user's activity level. Several higher order features from the raw sensor data stream are used to classify the activity level. Two aspects of the problem are considered, namely the accuracy of estimating the activity level as well as the power required in sensing and estimation. Experimental results on Android based smartphones demonstrate that an active user can get up to 37% increase in battery life without significant effect on the user experience.**

Keywords—Activity Recognition; Energy Efficiency; User Activity Modeling; Smartphone

I. Introduction

A smartphone is a mobile phone built on a mobile operating system, with higher computing capability and connectivity options compared to a feature phone. Inclusion of features such as accelerometers, GPS, compass, radio receivers, cameras and audio processors have made smartphones the primary device for personal assistance including communication, navigation, entertainment, imaging and information access. By far the most significant problem with most smartphones is their highly limited battery life. In recent times, hardware manufacturers have worked extensively to solve this problem, resulting in highly energy efficient processors and some promising directions in power-efficient displays. However, the wireless communication modules continue to consume significantly more energy in comparison and is a potential source for energy savings.

Today's internet is the mobile internet. People are moving towards smart-phones for general browsing, social networking and emails etc. Figure 2 shows the percentage of phone owners who used internet or email on their phone. 21% of smartphone users use their phones, not their PCs or tablets, as their main way to access the internet. Thus reducing the energy consumed by the network processes like Wi-Fi on smartphone devices could help in huge energy savings.

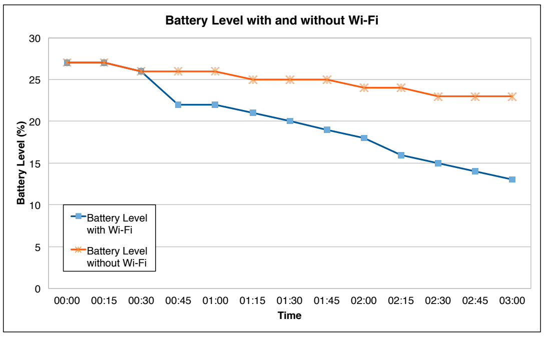

Fig. 1. Battery level variation on a smartphone with and without Wi-Fi enabled (Data transfer: 2KB at 5 minute intervals).

An interesting nature of wireless data communication arises the connectionless protocols used for data communication. The data is buffered at different points in the communication path and is delivered whenever a connection to the device is available. Moreover, most usage patterns involve intermittent periods of activity and inactivity. We take advantage of this fact to switch off the wireless data radio, whenever the usage is not significant. This would result in significant savings of power is used appropriately. However, there are two challenges that need to be addressed to achieve this. We need to be able to predict periods of activity and inactivity of the specific user to shut down the wireless appropriately. Inaccuracies in this prediction will result in a poor user experience or a lost opportunity for power saving. On the other hand the data collection for determining the user state and the classification process should be simple enough to avoid excessive energy usage for its purpose, nullifying any savings resulting from powering off the wireless.

In this work, we look at modeling and monitoring the user activity level (via machine learning) to decide when the wireless data modules should be turned off for maximal energy saving without compromising the user experience. The Wi-Fi scheduling is formulated as a classification problem, where Wi-Fi is scheduled based on the predicted usage of the user in the next time window. The specific problems that we deal with in this work may be summarized as:

- *Activity Modeling on Smartphone:* We propose a power-efficient approach to modeling the user's activity level on the smartphone device using various sensors and parameters that are easily available on most smartphones.

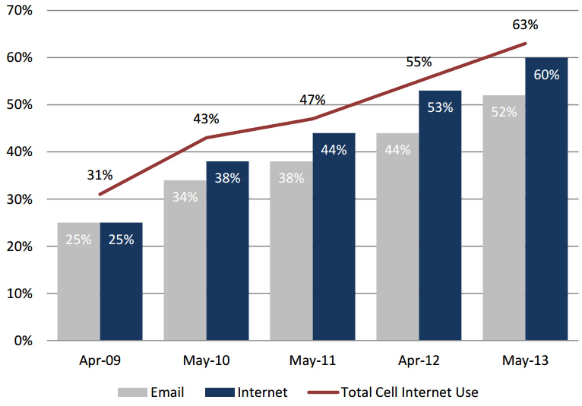

Fig. 2. Percentage of phone users who use internet or email on their phone in 2013 [1].

- *Energy Reduction by continuous Activity Scheduling:* We proposed an approach to schedule continuously running processes like Wi-Fi for better energy management without hampering the user experience.

We first take a look at the existing works in the area of activity classification and energy efficient mobile computing to understand the popular approaches in this area, which are then leveraged in coming up with an efficient way of activity modeling.

A. Related Work

Mobile phone based pattern recognition and data analysis has been a hot research field in the last decade. Specifically, the area of modeling user behavior to provide location, time and need based services has seen significant attention. Most smartphones are filled with a variety of applications that use sensors to measure movement and understand the world around us. They can be programmed to teach itself to work better, based on how you use it, the applications you run, and how you use it to communicate with others.

A lot of work has also been done on activity and gesture recognition with the help of accelerometer sensor data. Sun *et al.* [2] proposed a mechanism to classify physical activities of the bearer of the device using accelerometer data with varying positions and orientations. The embedded tri-axial accelerometer in a mobile phone can continuously log the experienced forces at each sampling interval and produce 3-D acceleration reading $A = (a_x, a_y, a_z)$ along three orthogonal axes. In addition to the raw sensor data, several features derived from the accelerometer sensor including mean, variance, correlation, energy, frequency-domain entropy, cepstral coefficients, and log-FFT frequency bands [3]–[10]. Further, Longstaff *et al.* [11] proposed different health related applications via semi-supervised activity modeling on smartphone. Realizing the effectiveness of accelerometer data in determining various usage parameters, mobile phone manufacturers have started embedding dedicated low-power motion sensor processors in addition to the main CPU on smartphones and tablets. In this work, we use accelerometer based features that were proved good by most of these methods for modeling the user activity. Along with accelerometer data, we also took other important indicators that suggest the type of user activity such as user interaction, and CPU and RAM usage on the device. A detailed description of the specific features that are used is given in Section IV-A.

Activity traces can improve the user interface mechanisms across a range of applications. For example, they can be used as automated reminder or triggered for an action or input at a convenient moment [12]. Recently, researchers have started applying various machine learning approaches to activity recognition on smartphones. Yan *et al.* [13] proposed an efficient continuous activity recognition system. They tried to model continuous user activity in a Markovian model based on the currently recognized activity. Zhuang *et al.* [14] looked at the problem of energy-efficient location sensing on a smartphone. They tried to find the location using substitution, suppression, piggybacking and adaptation methods, thus avoiding the use of power intensive sensors like GPS for getting the location. Unlike the above works that tries to do a specific task in an efficient manner, we use a combination of activity recognition and intelligent scheduling to save power dissipation by the most power hungry processes on a typical smartphone. Towards this, we present a more detailed analysis of the aspects of the phone that consumes the most amount of power (in Section II). This is followed by a description of our approach and the activity classification model. In Section V, we provide the results of various experiments comparing our approach with the base case of regular usage profiles.

II. What consumes Power?

Different applications have different power requirements. Figure 3 shows the relative power consumed by various services and applications on a smartphone device. It can be clearly observed that network services such as Wi-Fi, EDGE and 3G are most power consuming processes on the device, and of the three, Wi-Fi is the worst offender. However, in idle mode, Wi-Fi consumes very less power.

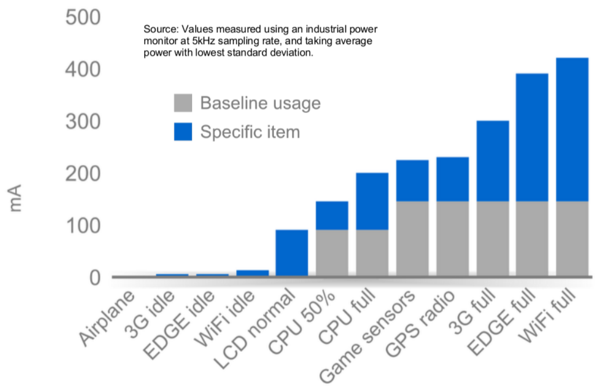
Source: Values measured using an industrial power monitor at 5kHz sampling rate, and taking average power with lowest standard deviation.

Fig. 3. Component-wise energy consumption on Android (Jeff Sharkey [15]).

The power consumed by the network services also varies with the duration between data transfer operations. Figure 4 shows the power consumed by the service when bundled transfer mode is used, compared to unbundled data transfer mode. In our experiments, the energy consumed by unbundled transfer was up to 3 times that consumed by bundled transfer.

The best practices for the network applications is to fetch as much data as possible in one go. Google recommends pre-

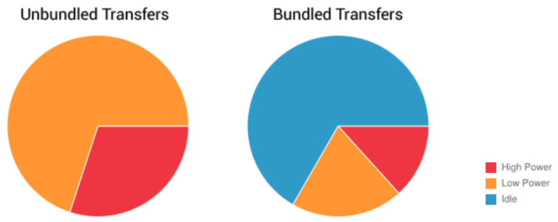

Fig. 4. Duration of time spend under different states (high power, low power and idle) by applications under Bundled and Unbundled packets transfer mode (Kumar Rangarajana [16])

fetching data that will initiate in next 2 to 5 minutes, and in order of 1-5 MB. So in order to save the battery, batch mode should be used rather than on-demand mode for sending and receiving the data. We tried to use this practice to improve the energy efficiency in the mobile device. The data is scheduled in batch mode by intelligently switching between the Wi-Fi states. Data is fetched in bulk only when it is required by the user and the Wi-Fi is turned off when user is sleeping or inactive on the mobile device.

III. Wi-Fi scheduling as Multi class Classification

Internet services are high power consuming processes. So they need to be scheduled in a personalized way in order to save battery. We model users behavior to schedule the Wi-Fi state over time. The user current activity level is predicted and the model is scheduled for the next half hour. If the user is highly active on the device, the Wi-Fi is turned on. Conversely, if the phone is idle the Wi-Fi is turned off to save the battery. This way the power is conserved without much hampering the users experience. There are three main modules. The feature extraction module, the classification module and the scheduling module. Figure 5 shows a flow diagram of our approach. The feature extraction module creates the feature vector using the accelerometer sensors, screen on duration and CPU and RAM usage by the device. The classification module learns these features and tries to predict the Wi-Fi usage by the user in next half an hour. The classification module interprets the user's activity state as one of the following classes:

1) *Very Low* (l_0) : Corresponds to no or very low activity by the user in the last half an hour.
2) *Low* (l_1) : Corresponds to low activity by the user in last half an hour.
3) *Medium* (l_2) : Corresponds to moderate activity usage by the user in last half an hour.
4) *High* (l_3) : Corresponds to intense activity by the user.

Based on the user state prediction, the Wi-Fi in the device is scheduled in the following manner.

- Label l_0 turn off the Wi-Fi for next half an hour.
- Label l_1 turns off the Wi-Fi for next 25 minutes. The Wi-Fi is then turned on for the next 5 minutes.
- Label l_2 turns off the Wi-Fi for the next 20 minutes. The Wi-Fi is then turned on for next 10 minutes.
- Label l_3 turns on the Wi-Fi for the next half an hour.

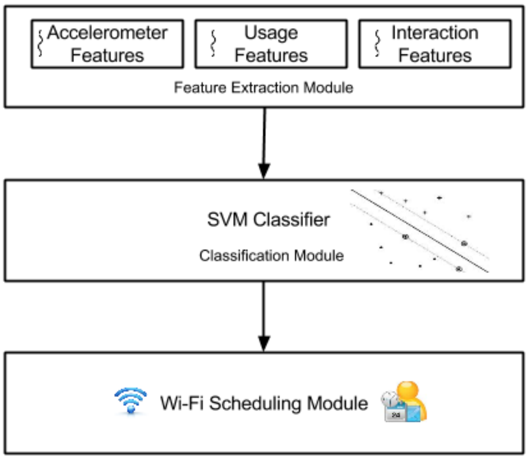

Fig. 5. Primary steps in the algorithm with data flow. Feature extraction module extracts the Accelerometer based, usage based and interaction based features. Classification module predicts the user activity level using SVM Classifier and Wi-Fi Scheduling Module schedules the Wi-Fi state based on the predicted output.

IV. Classification Frame Work

A. Feature Extraction

To extract the data from the smartphone, we created an Android application that records the required data. Data was collected over the period of half an hour to predict the users behavior for next cycle. After each half an hour, the application starts a service to collect the data. The service runs for the duration of 30 seconds and executes the following processes.

1) Partial wake-lock is acquired to prevent smartphone from going to sleep during the process execution.
2) The required data is collected for the duration of 30 seconds.
3) The user activity label is predicted by the classification module from the collected data.
4) The Wi-Fi is scheduled for next half an hour based on the predicted label.
5) Wake-lock is released and phone is allowed to go to sleep.

We divides the features into three categories- accelerometer based, system usage based and interaction based features. These features are explained in details below.

1) Accelerometer Based Features: Accelerometer based features are indication of the physical activity of the user like stationary, walking, running etc. The assumption behind is if the user is running or walking the probability of using his phone is much less as compare to stationary state.

Accelerometer continuously samples the acceleration at the specified sampling interval and produces 3-D acceleration readings $S = (S_X, S_Y, S_Z)$ along X, Y and Z direction. As the acceleration in a direction depends on the orientation of the phone, the overall acceleration $S_\phi = ||S_X, S_Y, S_Z||$ is also considered as feature. The sensor reading thus becomes a 4-D vector. Each device is calibrated differently. So we neglected the acceleration due to gravity while calculating the acceleration of the smartphone.

The accelerometer data is tracked for the user with the sampling frequency of $5Hz$. The accelerometer is tracked for 30 seconds and the following features are extracted.
Average mean along S_X, S_Y, S_Z and the S_ϕ vector.

$$\mu_d = \frac{\sum_{i=1}^{n} S_{d,i}}{n}; d \in \{X, Y, Z, \phi\}$$

Standard deviation along S_X,S_Y,S_Z and the S_ϕ vector. Standard deviation indicates the dispersion in the acceleration along different axis.

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^{n}(S_{d,i} - \mu_d)^2}{n-1}}; d \in \{X, Y, Z, \phi\}$$

Correlation among each pair of S_X, S_Y, S_Z and the S_ϕ vector. Correlation implies the strength between two corresponding axis signals.

$$r_{d,d'} = \frac{\sum_{i=1}^{n}(S_{d,i} - \mu_d) * (S_{d',i} - \mu_{d'})}{\sqrt{\sum_{i=1}^{n}(S_{d,i} - \mu_d)^2 * \sum_{i=1}^{n}(S_{d',i} - \mu_{d'})^2}};$$

$$d, d' \in \{X, Y, Z, \phi\}, d \neq d'$$

Fourier energy for S_X, S_Y, S_Z and the S_ϕ vector. The energy feature is the sum of all the squared DFT component magnitudes except the DC component of the signal as the DC component (mean) has been used as an individual feature. The magnitudes are divided by the number of components for normalization.

$$\varepsilon_d = \frac{1}{n}\sum_{i=1}^{n} ||\mathcal{F}(S_d)_i||; d \in \{X, Y, Z, \phi\}$$

Fourier entropy for S_X, S_Y, S_Z and the S_ϕ vector. Fourier entropy is the normalized information present in the DFT of the signal, excluding the DC component.

$$\delta_d = \sum_{i=1}^{n} ||\mathcal{F}(S_d)_i|| * log(\frac{1}{||\mathcal{F}(S_d)_i||})$$

The intuition behind using accelerometer based features is that Wi-Fi requirements are different during different activities of the user.

2) System Usage Based Features: System Usage based features tells how extensively the phone is being utilized. Features will be higher if the phone is used for heavy applications like voice calling, network browsing vs. when the phone is used for viewing SMS, setting alarm etc.

The system usage feature of the smartphone is computed based on the following entity.

- The average CPU utilization of the smartphone.
- The average memory used by the applications in the smartphone.

The CPU utilization tells about state of running applications and the memory used tells about the state of installed applications and background services.

3) Interaction Based Features: Interaction based features tells how frequently user is using his phone. More frequent the user is on the device, more likely he will requiring the network connectivity.
Interaction based features consist of the following features.

- *Screen On Time*: In half an hour duration, total time in milliseconds for which screen was on.
- *Frequency Of Interactions*: In half an hour duration, total number of interaction (that is number of time the screen has been turned off and turned on) by the user.

The features are extracted in parallel so as to get the consistent data and reduce the battery consumption due to wake-lock acquisition by the application. Finally we concatenate all the features to form a final feature vector for the classification.

B. Training and Classification

The user's activity is modeled based on his behavior. The features extracted by the feature extraction module is fed to the classification module for prediction. We classified the user's activity level into four different classes, labeled by l_0, l_1, l_2 and l_3.

1) Training: For training purposes, we created an application that logs the accelerometer, screen on duration,internet, CPU and RAM usage of the device at every half an hour interval. We scaled the internet usage based on the screen on duration in half an hour as well as and the data traffic. The assumption is if the screen is off and the Wi-Fi is still on then without any data traffic this should be handled in intelligent way. We have categorized user into four categories no usage, low usage, medium usage and high usage. Class label of a feature vector created during current time window is decided based on the internet usage by the user in next time window. We run the application for 6 working days which resulted into 310 feature vector. For preserving the usability of the application, the classes are given different penalty scores. Table I shows the weights assigned to different classes while training.

TABLE I. CLASS WEIGHTS ASSIGNED TO DIFFERENT ACTIVITY LEVEL WHILE TRAINING.

Class Label	Very Low (l_0)	Low (l_1)	Medium (l_2)	High (l_3)
Weight	1.0	2.71	8.92	4.92

2) Classification: For classification, the trained SVM model along with the feature extraction module is deployed on the smartphone. The features are collected for half an hour interval. After half an hour, the prediction (based on the features from feature extraction module) is done. The SVM model created during training is used for predicting the labels. The predicted label is then send to the Wi-Fi scheduling module and the Wi-Fi is scheduled appropriately.

V. Experimentation and Results

We evaluated our approach based on two metrics, the power saved and the user happiness. Power saved metric measures the overall power saved by our method. Although the power saving is important, it should not hamper the user's overall experience on the smartphone. So user happiness testing is performed to test the user's satisfaction with the software.

For measuring the power saved by our software, we subjected the phones to real-world behaviors (via calls, messages and emails) under two different settings, when our proposed method is installed and when it is not. The percentage drop in the power is computed in both the settings to find the energy efficiency achieved by our method. For finding the user happiness, we installed the application on smartphones of two different users and recorded the number of times they switched the Wi-Fi state manually. Each manual switch is considered as a penalty on user happiness. Section V-A explains the implementation of the system. Section V-B shows the power saved by the application and Section V-C shows the user happiness scores obtained when the application was installed on user's smartphone.

A. Implementation

An android application is created for capturing the features, predicting the activity and scheduling the Wi-Fi. The application schedules a prediction service at intervals of half an hour. Along with the service, it also registers a broadcast receiver that listens to screen state change event and logs the changes. After every half an hour, the service wakes the android system and 3D accelerometer readings are recorded for 30 seconds. Based on the readings, the accelerometer signal's mean, standard deviation, correlation and Fourier transform is computed. Along with these features, the screen on time and frequency of interactions are also evaluated using the logs created by the broadcast receiver. Android OS logs the internet, CPU and memory usage of the system. The average internet, CPU and memory usage is calculated based on the logs created by the OS. To maintain consistency among different features, all the features are extracted at the same time. Features are concatenated into a single feature vector. SVM classifier is used to predict the usage in next half an hour (using a pre-trained model) utilizing these features. Based on the predicted output by the classifier (high, medium, low or very low), the Wi-Fi is turned off and an alarm manager is scheduled to turn it on after suitable interval of time. All the decisions made by the classifier are logged for the purpose of evaluation. For evaluating user happiness, all Wi-Fi state changes are also logged. The number of manual state change events are extracted and considered as a measure for calculating user happiness. The activity prediction is shown to be achievable using light-weight computations without consuming much power or memory. Our application took only 5MB of RAM and 101KB external memory, which is similar to the average requirements for an application on smartphones. Also, our application took 1% of battery in 24 hours[1].

B. Power saved by the software

User have different activities scheduled at different times. For example, number/durations of calls in one hour might be very different from the number of calls in the next hour. Similarly the power consumed in other tasks like messages, games etc. might be different in the two hours. So, calculating the battery drop in the two corresponding hours (once with the scheduler and once without the scheduler) might not be an accurate measure to evaluate the power saved by our software.

To evaluate the power saving by our software we took two device from both the categories (refer Table III for specification) having same set of application, battery state, memory usage etc., and performed below test plan on all four devices, two having our software installed (one from each category) and two without our software. Table II lists the activities carried out on the smartphone. All the activities are performed at almost same time and for same duration. For example, the calls on all the devices are performed simultaneously for 2 minutes duration. Similarly, all the messages and emails are also send simultaneously so that the loss incurred by other processes remains same. The controlled experiment was performed for 4 hours in similar settings and the battery dissipated on each device is recorded. Figure 6 shows the relative percentage drop in battery on the devices after each hour.

TABLE II. ENERGY EFFICIENCY TESTING: PERCENTAGE BATTERY DROPPED IN SAMSUNG GALAXY S (DEV.1) AND HTC EXPLORER (DEV.2) DURING CONTROLLED TESTS.

	Messages Received	Number of Calls	E-Mails Received and Replied	Drop in Battery (Dev.1)	Drop in Battery (Dev.2)
Without Scheduler	25	15	25	38%	52%
With Scheduler	25	15	25	28%	15%

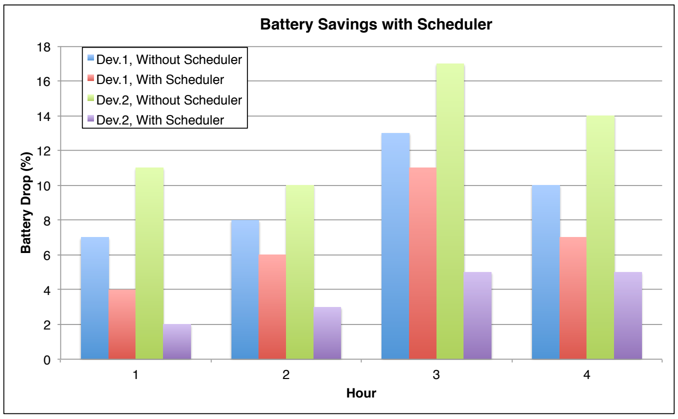

Fig. 6. Percentage drop in battery with time under different settings (Dev.1 and Dev.2).

C. Measuring User Happiness

Power saving is important but at the same time we have to take care that user experience should not be hampered a lot. To verify this we conducted an experiment in which we have given our software to users and logged there battery usage in two categories first when the user is very active second when the user is less active. Along with battery usage, the number of times the Wi-Fi state is changed manually is also recorded. Each Wi-Fi switch by the user is considered as misclassification by the algorithm. The higher the number of switches by the user, the more unhappy the user is with the application. Table IV shows 6 hrs analysis when the user is highly active (working hours) and Table V shows the 6 hrs analysis when the user is not much active (non-working hours) on the smartphone device.

Our application took just 1% of battery in 24 hours and saved 10 to 37% of battery in 4 hours without much affecting

[1] The power consumed by the application is recorded through battery info manager on android OS.

TABLE III. SPECIFICATIONS OF THE PHONES ON WHICH TESTING WAS CONDUCTED.

	Feature	Dev.1: Samsung (Galaxy S)	Dev.2: HTC (Explorer)
General	[2G Freq], [3G Freq], [SIM]	[GSM 850/900/1800/1900], [HSDPA 900/1900/2100], [Mini]	[GSM 850/900/1800/1900], [HSDPA 900/1900/2100], [Mini]
Display	[Type], [Size]	[AMOLED capacitive touch, 16M colors] , [480x800, 4.0"]	[TFT capacitive touch, 256K colors] , [320x480, 3.2"]
Memory	Internal	8/16 GB storage, 512 MB RAM, 512 MB ROM	90 MB storage, 512 MB RAM, 512 MB ROM
Data	[GPRS], [EDGE], [Speed], [WLAN], [Bluetooth], [USB]	[Class 12 (4+1/3+2/2+3/1+4 slots), 32 - 48 kbps], [Class 12], [HSDPA, 14.4 Mbps; HSUPA, 5.76 Mbps], [Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot], [Yes, v3.0 with A2DP], [Yes, microUSB v2.0]	[Up to 80 kbps], [Up to 236.8 kbps], [HSDPA, 14.4 Mbps; HSUPA, 5.76 Mbps], [Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot], [Yes, v3.0 with A2DP,EDR], [Yes, microUSB v2.0]
Features	[OS], [Chipset], [CPU], [GPU], [Sensors], [Messaging]	[Android OS, v2.3 (Gingerbread)], [Qualcomm MSM8255T Snapdragon], [1.4 GHz Scorpion], [Adreno 205], [Accelerometer, proximity, compass],[SMS(threaded view), MMS, Email, Push Mail, IM, RSS]	[Android OS, v2.3 (Gingerbread)], [Qualcomm MSM8255T Snapdragon], [600 MHz Cortex A5], [Adreno 200], [Accelerometer, proximity],[SMS(threaded view), MMS, Email, Push Mail, IM, RSS]

TABLE IV. USER HAPPINESS (WORKING HOURS): COUNT OF NUMBER OF TIMES WI-FI WAS MANUALLY TURNED ON.

	Wi-Fi turn-on count (Usr 1)	% Battery Drop (Usr 1)	Wi-Fi turn-on count (Usr 2)	% Battery Drop (Usr 2)
Without Scheduler	NA	38	NA	33
With Scheduler	5	29	3	13

TABLE V. USER HAPPINESS (NON-WORKING HOURS): COUNT OF NUMBER OF TIMES WI-FI WAS MANUALLY TURNED ON.

	Wi-Fi turn-on count (Usr 1)	% Battery Drop (Usr 1)	Wi-Fi turn-on count (Usr 2)	% Battery Drop (Usr 2)
Without Scheduler	NA	26	NA	8
With Scheduler	2	19	0	7

user convenience. The battery usage vs. amount of battery saved confirms that the application can be used non-stop and can save significant amount of power, especially for heavy internet users.

VI. CONCLUSIONS AND FUTURE WORK

Power consumption and the consequent low battery life is one of the primary limiting factors for modern mobile devices. In this work, we proposed an effective way to nullify some of this problem by predicting user activity levels and scheduling the power hungry wireless modules accordingly. The encouraging results of the experiments lead us to affirm that a step forward has been taken to personalize the equipment so that it can function well while improving power efficiency.

The effectiveness of the proposed approach can be improved and extended in many ways. Application time interval can be scheduled dynamically, making it adaptive to user's activity level, which should give better result in terms of user happiness as well as battery saving. As the application is monitoring and modeling the user's activity, it can also perform additional tasks such as silencing the phone between meeting/sleep hours, or put the phone in silent mode when the user is driving.

REFERENCES

[1] M. Duggan and A. Smith, "Cell internet use 2013," 2013, http://pewinternet.org/%7E/media//Files/Reports/2013/PIP%5FCellInternetUse2013.pdf.

[2] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 548–562.

[3] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Pervasive Computing*, vol. 3001, pp. 1–17, 2004.

[4] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 766–772.

[5] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, ser. IAAI'05. AAAI Press, 2005, pp. 1541–1546.

[6] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, Washington, DC, USA, 2006, pp. 113–116.

[7] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proceedings of the 4th International Conference on Pervasive Computing*, 2006, pp. 1–16.

[8] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-d accelerometer," in *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 25–38.

[9] A. Rai, Z. Yan, D. Chakraborty, T. K. Wijaya, and K. Aberer, "Mining complex activities in the wild via a single smartphone accelerometer," in *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data*, 2012, pp. 43–51.

[10] G. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Activity Context Representation: Techniques and Languages Workshop at Association for the Advancement of Artificial Intelligence*, Toronto, CN, 2012.

[11] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning." in *In Proceedings of Pervasive Health*. IEEE, 2010, pp. 1–7.

[12] A. Sixsmith and N. Johnson, "A Smart Sensor to Detect the Falls of the Elderly," *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, vol. 3, no. 2, pp. 42–47, 2004.

[13] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach." in *Proceedings of 16th Annual International Symposium on Wearable Computers (ISWC)*. IEEE, 2012, pp. 17–24.

[14] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 315–330.

[15] J. Sharkey, "Coding for life battery life, that is," 2009, http://dl.google.com/io/2009/pres/W%5F0300%5FCodingforLife-BatteryLifeThatIs.pdf.

[16] K. Rangarajana, "Android battery consumption - the what and how," 2012, http://www.slideshare.net/littleeye/power-optimization-for-android-developers.

[17] C. Shore, "Developing power-efficient software systems on arm platforms," 2010, http://www.iqmagazineonline.com/current/pdf/Pg48-53.pdf.