Activity Recognition on Smartphones: Efficient Sampling Rates and Window Sizes

Baris Yamansavascilar
Department of Computer Science
Yıldız Technical University
Istanbul, TURKEY
l1110001@std.yildiz.edu.tr

M. Amac Guvensan
Department of Computer Science
Yıldız Technical University
Istanbul, TURKEY
amac@ce.yildiz.edu.tr

***Abstract*— Great hardware and software capabilities of mobile devices allow us to research new scientific fields. Activity recognition is one of the main research areas for smartphones. Built-in sensors of a standard smartphone, such as accelerometer, magnetometer, gyroscope, enable us to recognize the daily activities of a person. In this study, we focused on the window sizes and the sampling rates in order to observe how they affect to the accuracy and CPU utilization. For our test scenarios, we built a dataset including a tri-axial accelerometer sensor data of 7 daily activities: *walking, jogging, sitting, lying, standing, walking upstairs* and *walking downstairs*. We collected these activities with a sampling rate of 80 Hz by using 5 seconds window size. Then, we downsampled the collected data to 40 Hz, 20 Hz, 10 Hz, 5 Hz and 1 Hz by using 4, 3, 2, and 1 seconds window sizes, respectively. Thus, we could evaluate the variation of the accuracy and CPU utilization. Our test results showed that when the sampling rate increases, both the accuracy and the CPU utilization become greater. Moreover, we observed that, for a fixed sampling rate, when window size increases, CPU utilization decreases.**

Keywords*— *Activity Recognition, Sampling Rate, Window Size, CPU utilization, Classification Accuracy, Smartphone, iOS

I. INTRODUCTION

In recent years, smart mobile devices have entered into every part of our lives and they have changed our life styles. Their great hardware and software capabilities have facilitated our lives with various ways such as sending e-mail, browsing on the Internet, reading books and magazines, playing games etc. Especially, after the announcement of the original iPhone in 2007, the usage rate of the smartphones has accelerated. This acceleration has changed computer industry and routines of people. One of the most important differences between smartphones and desktop-notebook computers is the sensing capability, which turns the mobile phones into a great assistant of a person. With the help of the sensors, smartphones ease our lives in many ways such as health assistance, detecting falls, prolonging the battery life, finding a location, protecting our privacy. Recently, researchers have focused on mobile applications especially exploiting sensor data. Thus, activity recognition systems have become popular for the last 5 years. Most of researchers use accelerometer, magnetometer and/or gyroscope sensors of the smartphone to recognize daily activities of a person.

Activity recognition systems aim at recognizing daily movements of people and reporting them. Generally, these systems are designed to make people be aware of their health and fitness performance. For example, obese people care about their daily movements and calorie consumption. Besides, exercises are important for both elderly people and athletes to maintain their body health. Researchers take the outputs of the activity recognition system and create several reports to inform the user.

Before the spread of smartphones, activity recognition systems were implemented by using wearable sensors that were placed on human body [1]. These wearable sensors might have one or two-dimensional accelerometer in contrast to smartphones, which usually have 3-dimensional accelerometer. Moreover, mobility and simple usability of smartphones have encouraged researchers to exploit the smartphones for activity recognition. Nowadays, most of the activity recognition systems are implemented on smartphones or other smart mobile devices such as fitness bands and watches. Thus, in this study, we implemented a mobile application on a smartphone, iPhone 4, for our test scenarios. Moreover, we validated our test results using an iPod Touch 6th Generation.

One of the main problems of smartphones is the battery life. Even though most manufacturers have focused on this problem, battery of the smartphones still runs out usually within 1 - 1.5 days regarding a standard usage. Especially, battery life is more important for sensor applications such as activity recognition, fall detection and sleep analysis. Since these applications frequently read data from built-in sensors within milliseconds, they use CPU much more than other applications. Regarding to this fact, we make the following contributions: (1) observation of the influence of window sizes to the CPU utilization; (2) determination of the effective window sizes and sampling rates by taking into consideration both the CPU utilization and the accuracy. Thus, it will lead us to infer the relation between *sampling rate, window size, CPU utilization* and *accuracy*. In this study, since the results of CPU utilization could be generalized for different kind of mobile phones more accurately, we used these results rather than the power consumption which depends much more on the manufacturer, memory unit, operating system, sensors, disk I/O etc.

For our test scenarios, we implemented a mobile application for activity recognition in order to collect data of daily activities. In addition to our application, we exploit a module, which logs the CPU utilization in order to evaluate the influence of the window sizes and sampling rates. Our mobile application is designed also to calculate calorie consumption of the user depending on his/her activities. We used Metabolic Equivalents (MET) [18] to calculate calorie consumption.

The rest of the paper is organized as follows. In Section II, we introduce related studies and their methods. In Section III, we describe the design of our activity recognition system. In Section 4, experimental results are discussed in detail. Finally, we conclude the paper in Section 5.

II. Related Works

Activity recognition based on accelerometer data has taken much attention in the past decade. One of the earliest studies in this topic was performed by Bao & Intille [1]. They placed wearable biaxial accelerometers on five different part of human body. In their study, 20 daily activities were studied by using 76.25 Hz sampling rate and each window represented as 6.7 seconds. Also, decision tree classifiers showed the best performance. In [2], Abdullah et al. discussed classification algorithms and evaluation methods of smartphone-based human activity recognition. They stated that the selection of classification algorithm is based on the capability of the processing platform to execute the algorithm. In addition, they grouped all features used by researchers into four categories as magnitude-based, frequency-based, correlation and others. In [9], Kwapisz et al. implemented a mobile application on Android. Their study is one of the earliest studies that used a commercial mass-marketed device rather than a research-only device. They studied six daily activities of twenty-nine users. Also, they used ten-fold cross validation by using 10-seconds interval and 20Hz sampling rate. In [10], Anjum & Ilyas built a smartphone application which tracks users' activities and reports estimates of the burned calories. C4.5 classification algorithm was the most successful algorithm with 95.2% of accuracy. In this study, MET was used to compute calorie consumption as we used in our application. On the other hand, some studies [11,12,13,14] focused on the position and orientation of the smartphone. They proposed orientation and position independent methods for activity recognition. Moreover, some studies [26, 27] worked on fall detection, which is a part of activity recognition.

Some works studied both on complex and simple activities. In [3], Dernbach et al. divided activities into two categories as simple and complex. While complex activities were defined as cleaning, cooking, medication, sweeping, washing hands and watering plants, basic activities were defined as biking, climbing stairs, driving, lying, running, sitting, standing and walking. They used fastest threshold mode of the smartphone to collect data. Also, six classifiers were tested on Weka [20] including Multi-layer Perceptron, Naïve Bayes, Bayesian network, Decision Table, Best-First Tree, and K-star. Accuracy of algorithms was over 90% for basic activities while 50% for complex ones. In [4], Do et al. aimed to maintain body health of the user. They studied on basic and complex activities. Tri-axial accelerometer of the smartphone was exploited by using 5Hz-sampling rate. In their study, activity recognition process was carried out on a web server. In [5], Rai et al. evaluated an unsupervised clustering based approach to perform complex activity recognition. In their study, users were permitted to tag their activities with no restriction.

There are also different studies. In [6], Kwapisz et al. identify/authenticate the user of smartphone by using his/her physical activities. They collected accelerometer data from 36 users and used 10 seconds interval with 20 Hz sampling rate. In [7], Weiss & Lockhart extended the aforementioned study [6] and they used identification of user to recognize user characteristics. In [8], Kim et al. aim at detecting the early symptoms of dementia. They compared current and stored activity patterns of the user to infer dementia by using activity pattern matching. They placed wearable tri-axial accelerometers on six different part of human body.

In literature, to the best of our knowledge, there are no studies that investigate the accuracy, CPU utilization, sampling rate and window size together. Available studies examined only two or three of these factors. In [15], Lau & David investigated which sampling rate and window size combination of feature extraction will provide better activity recognition using smartphones. They used Nokia N95 as smartphone. Test results showed that sampling rate of 10 Hz and 20 Hz is sufficient to achieve good accuracy. Also, the combination of sampling rates with window sizes of 2 and 4 seconds gave higher accuracies than smaller window sizes. In [16], Anguita et al. proposed a novel hardware-friendly approach, since mobile phones are limited in terms of energy and computing power. Their new method adapted standard SVM and they used integer parameters to reduce computational cost. In their study, while accuracy of standard SVM is 89.3%, their new SVM model achieved 89% of accuracy. In [17], Yan et al. investigated how the sampling frequency and classification affect each activity separately. They focused on two independent parameters: the sensor sampling frequency and the set of features. To reduce energy overheads, they studied the combined influence of these two parameters on the accuracy, separately for each activity. They achieved 50% of energy saving under ideal conditions. On the other hand, energy saving on the phone was between 20-25%. In [18], Viet et al. proposed an adaptive energy saving strategy choosing appropriate sampling frequency & classification features for each activity, similarly to [17]. They contributed a novel method for feature extraction and used SVM as classifier. They achieved 28% of energy saving on mobile phone.

Our study differs from existing works in the following ways.

- We observed the influence of window sizes to the CPU utilization while other studies measured the power consumption by considering only the effect of sampling rates or features.
- We investigated the effective window sizes and sampling rates by analyzing the results of both CPU utilization and the accuracy whereas other studies examined only two or three of these factors.

III. System Design

In this study, we evaluated the effect of different sampling rates and window sizes on the accuracy and CPU utilization. To perform this purpose, we first implemented an activity recognition system on a smartphone properly. Thus, we followed the classical aspects given in Fig. 1.

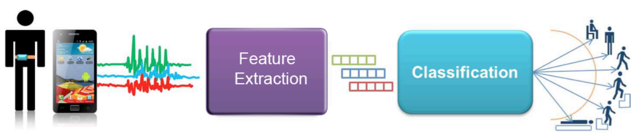

Fig. 1. Activity Recognition Task [16]

Training phase is a necessary step for an activity recognition system that uses supervised learning methods. For this purpose, we developed a module in our mobile application to collect raw sensor data and then labeled them. With the help of this module, accelerometer data could be collected and saved into mobile phone's file system. After obtaining raw data with a sampling rate of 80 Hz by using 5 seconds window size, the data was downsampled to create new datasets of 40 Hz, 20 Hz, 10 Hz, 5 Hz and 1 Hz. We recurred this data-preprocessing step four times to evaluate window sizes of 4 seconds, 3 seconds, 2 seconds, and 1 second. Thus, we achieved the data set that was used for accuracy measurements. The numbers of instances in the dataset based on the window sizes are given in Table 1. Afterwards, features were extracted according to the sampling rates and window sizes, separately. Finally, these feature-vector sets were given to the Weka tool [20] for the evaluation of classification algorithms.

After the evaluation of the accuracy, CPU utilization was examined by using the accuracy results, window sizes and sampling rates. For this purpose, we modified our earlier module to save the CPU utilization ratios by considering each <Window Size, Sampling Rate> tuple. To interpret the output, we exploited the Instruments tool.

In the following subsections, we explain the data collection and feature extraction in detail.

A. Data Collection

In this study, we considered 7 main activities for the activity recognition. These activities include *walking, jogging, sitting, lying, standing, walking upstairs* and *walking downstairs*. Three volunteers with different physical characteristics (1 overweight person, 1 tall person and 1 short person) helped us to collect data for each activity. During data collection, these three subjects carried the smartphone in the front pocket of their trousers at vertical position. Each subject performed each activity - except walking upstairs/downstairs - for two minutes.

Figure 2 shows the two screenshots of the training phase module. In this module, subjects first select the current activity they perform and enter their name. Afterwards, subjects hit the start button and record raw sensor data of their current activity. Labeling is performed automatically by using the information of the selected activity.

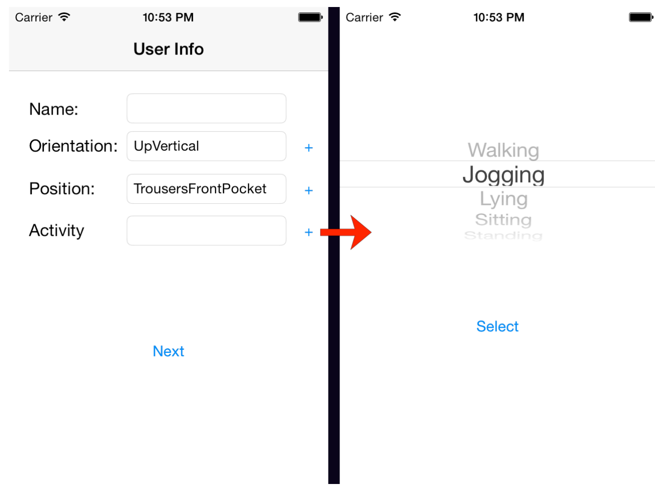

Fig. 2. User Interfaces For Collecting Raw Data

After recording raw data, we run the same module to save the CPU utilization ratios. As we want to evaluate the result of each <Window Size, Sampling Rate> tuple, we modified the module for each experiment. Since iOS operating system allows developers to log hardware usage rates automatically, we did not use any special methods.

B. Feature Extraction

Raw time-series accelerometer data cannot be given into standard classification algorithms [9]. It first must be processed and transformed into instances. To achieve this, we divided our labeled raw data into 4 seconds, 3 seconds, 2 seconds, and 1-second samples (windows). Afterwards, 61 time-domain features were calculated for each instance in the dataset using statistical methods.

IV. Experimental Results

In this study, we performed our experiments on Weka and Instruments, which is a performance-analysis and testing tool for dynamically tracing and profiling OS X and iOS code [24]. In

Instruments, Energy Diagnostic instrument was used to observe CPU utilization [25]. We evaluated six different classification algorithms including *J48 (C4.5), k-Star, Naive Bayes, Bayes Net, Random Forest, and k-NN*. We applied 10-fold cross validation method in all test operations.

TABLE I. THE NUMBERS OF INSTANCES FOR EACH WINDOW SIZE

	Window Size (Seconds)				
Activity	**1**	**2**	**3**	**4**	**5**
Walking	535	267	177	133	107
Jogging	510	254	169	126	102
Sitting	520	259	172	128	104
Lying	550	275	183	137	110
Standing	530	264	175	131	106
W. Upstairs	120	60	39	29	24
W. Downstairs	95	47	31	23	19

To evaluate CPU utilization on the Instruments by considering <Window Size, Sampling Rate> tuple, we run our module on iPhone 4 operating by Apple A4 processor. Apple A4 processor is based on the ARM processor architecture [21]. It combines an ARM Cortex-A8 CPU with a PowerVR GPU [22, 23]. Its maximum CPU clock rate is 800Mhz on iPhone 4. After we achieved our test results, we validated them using an iPod Touch 6th Generation.

A. Effects of the Sampling Rate and Window Size on Accuracy

To evaluate the effects of the sampling rates and window sizes on accuracy, we assessed the results of six classification algorithms shown in Table 2. In this Table, we divided the results in sections by using the sampling rates. Each group of the same sampling rate indicates a section. In each section, we underlined the best result for each algorithm. Moreover, we colored the best results (by considering all sections) for each algorithm as red. Analyzing the results considering the sampling rates show us that higher rates provide better accuracy. However, some exceptions have been occurred. For example, the accuracy of J48 for <4, 40> is higher than <4,80> whereas we expect the against.

Table 2 helps us to make the following inferences about window sizes. First, we observed that best results were achieved with the window sizes in the range of 3 and 5 seconds. Second, we noticed that generally when the sampling rate decreases, bigger window sizes give better accuracy.

The results of J48 algorithm are illustrated in Figure 3 in order to show the relationship between sampling rate, window size and accuracy more clearly.

TABLE II. ACCURACY OF ALGORITHMS BASED ON WINDOW SIZES AND SAMPLING RATES

WS (Sec)	SR (Hz)	J48	RF	k-Star	Naive Bayes	Bayes Net	k-NN
1	80	93.99%	94.65%	94.90%	87.62%	93.39%	94.16%
2	80	95.16%	95.93%	92.85%	88.08%	93.83%	94.53%
3	80	94.93%	**96.19%**	**96.09%**	**89.32%**	**94.08%**	**95.88%**
4	80	94.91%	**96.18%**	93.21%	88.97%	**94.06%**	**95.47%**
5	80	**95.98%**	**96.68%**	95.80%	**89.16%**	93.88%	93.71%
1	40	93.43%	95.24%	94.83%	87.13%	93.18%	93.95%
2	40	94.74%	95.51%	94.53%	87.03%	93.48%	94.88%
3	40	95.45%	95.77%	95.24%	**89.22%**	**94.39%**	94.82%
4	40	**96.04%**	**96.04%**	95.05%	88.40%	93.35%	94.63%
5	40	94.93%	**96.33%**	**95.28%**	88.46%	93.53%	**95.45%**
1	20	92.66%	94.58%	93.88%	83.85%	92.45%	92.62%
2	20	93.97%	94.53%	93.55%	84.43%	**93.41%**	92.99%
3	20	94.19%	95.03%	**95.45%**	86.26%	92.81%	94.08%
4	20	**95.62%**	**95.90%**	94.77%	86.14%	92.93%	93.92%
5	20	94.23%	94.93%	94.41%	**87.76%**	92.83%	**94.23%**
1	10	90.59%	92.69%	91.19%	80.04%	90.31%	90.31%
2	10	92.57%	93.48%	92.77%	81.77%	91.58%	92.36%
3	10	**93.02%**	93.55%	93.66%	82.98%	91.54%	91.86%
4	10	92.08%	94.20%	**94.34%**	84.16%	**92.22%**	**93.21%**
5	10	92.48%	**94.41%**	92.66%	**84.79%**	91.96%	90.56%
1	5	88.60%	91.15%	88.53%	75.66%	87.52%	86.96%
2	5	89.69%	92.36%	90.46%	78.40%	91.10%	90.11%
3	5	90.38%	92.81%	91.75%	80.44%	90.17%	89.96%
4	5	90.66%	93.91%	**92.65%**	81.47%	90.81%	89.96%
5	5	**91.61%**	**94.76%**	92.31%	**83.22%**	**91.78%**	**91.43%**
1	1	80.00%	82.76%	77.62%	63.60%	79.72%	79.55%
2	1	81.07%	85.34%	75.60%	71.46%	83.59%	74.82%
3	1	82.56%	**88.05%**	81.50%	74.95%	83.62%	80.66%
4	1	**84.87%**	87.41%	81.61%	76.66%	84.44%	81.05%
5	1	79.72%	86.54%	**83.39%**	**77.10%**	**84.62%**	**81.64%**

B. Effects of the Sampling Rate and Window Size on CPU Utilization

We organized CPU utilization records for each <Window Size, Sampling Rate> tuple into the line-based charts in order to evaluate the results more accurately. Fig. 4 and Fig. 5 show these results based on the window sizes and sampling rates, respectively. For varying window sizes, we observed that CPU utilization decreases when window size increases. However, after the window size of 3 seconds, the deceleration of the CPU utilization is considerably low. Thus, it can be negligible. Since the highest accuracy for each classification algorithm is obtained between 3 and 5 seconds window sizes, this observation is quite important.

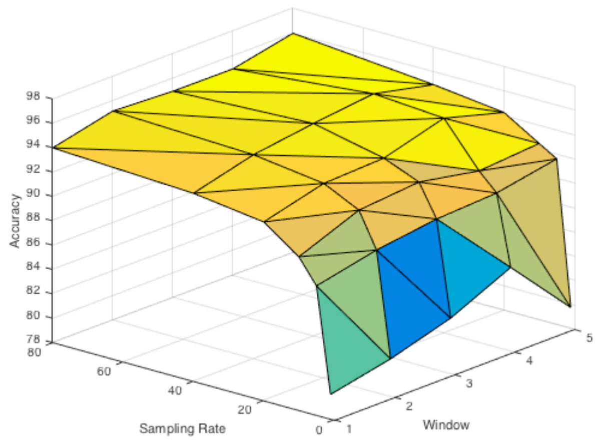

Fig. 3. The relationship between sampling rate (in Hz), window size (in seconds) and accuracy (in %) for the results of J48

Evaluating both sampling rate and window size shows us that the influence of the sampling rate on CPU utilization is bigger than the window size. For each transition from lower sampling rate to higher sampling rate, CPU utilization increases with a growing acceleration as given in Table 3. Note that, the transition between 1 Hz and 5 Hz is five times, whereas other transitions are two times.

C. *Efficient Sampling Rates and Window Sizes*

To determine efficient sampling rates and window sizes, we considered both CPU utilization and accuracy results. We must ensure the maximum benefit from accuracy and the minimum loss from CPU utilization. Thus, we carried out our assessments according to this purpose.

In our assessments, we first focused on window sizes. From the view of accuracy aspect, classification algorithms reached their peak when we have used the window sizes in the range of 3 and 5 seconds. Besides, we observed that generally when sampling rate decreases, bigger window sizes give better accuracy. From the view of CPU utilization aspect, when the window size increases, the CPU utilization decreases. However, especially after the window size of 3 seconds, this deceleration is considerably low. Thus, we can say that it can be ignorable. Finally, since CPU utilization and accuracy results indicate the same range, we concluded that the efficient window size is in the range of 3 and 5 seconds.

TABLE III. ACCELERATION VALUES OF EACH TRANSITION BETWEEN SAMPLING RATES FOR CPU UTILIZATION

	Window Size				
Transition	**1 sec**	**2 sec**	**3 sec**	**4 sec**	**5 sec**
1 Hz – 5 Hz	2.14	2.06	2.10	2.31	2.24
5 Hz – 10 Hz	1.41	1.41	1.59	1.47	1.61
10 Hz – 20 Hz	1.48	1.62	1.63	1.70	1.70
20 Hz – 40 Hz	1.60	1.62	1.68	1.83	1.81
40 Hz – 80 Hz	1.66	1.80	1.78	1.79	1.82

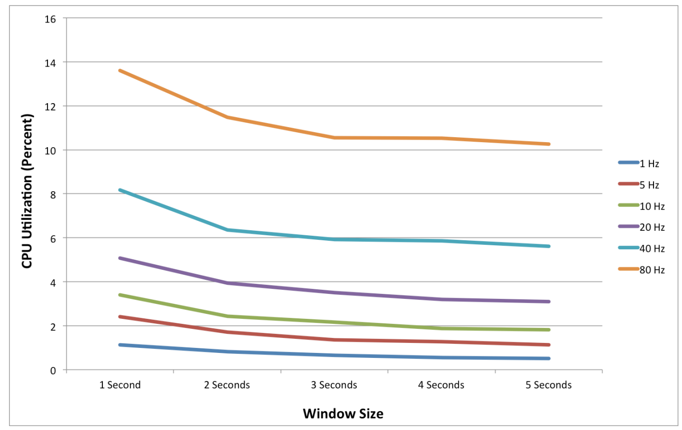

Fig. 4. CPU utilization based on window size

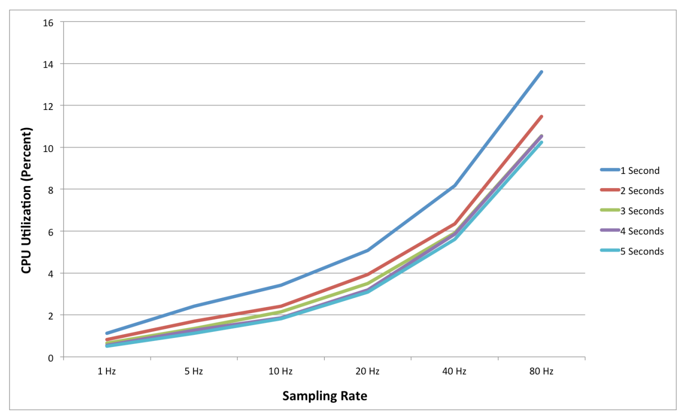

Fig. 5. CPU utilization based on sampling rate

According to our results, the influence of the sampling rates is much more than the effect of the window sizes. We observed that the CPU utilization increases with a growing acceleration when the sampling rate rises. Based on this outcome, we can say that the CPU utilization is inadequate to find the efficient range for sampling rate. Thus, we should consider both accuracy and CPU utilization together in order to make a conclusion about the efficient sampling rates. In our accuracy results, in most cases, we noticed that the sampling rate of 5 Hz is sufficient for a good accuracy (>90%). Likewise, for high accuracy (>95%), the sampling rate of 20 Hz is sufficient in most cases. If we consider the fact that the CPU utilization increases for higher values of the sampling rates, there should be a balance between them. However, since each algorithm have different characteristic, the optimum sampling rate (the balance) varies. Thus, we concluded that the efficient sampling rate is in the range of 5 Hz and 20 Hz.

V. CONCLUSION

In this study, we investigated the efficient sampling rates and window sizes for activity recognition on smartphones. Activity recognition is a multi-objective optimization problem. To simplify this problem we preferred to give our results within ranges instead of indicating an optimal point. For sampling rates, we concluded that the sampling rate in the range of 5 Hz and 20 Hz is the efficient in most cases considering both accuracy and CPU utilization. Likewise, for window sizes, we observed that the efficient size is in the range of 3 and 5 seconds. Moreover, for each <Window Size, Sampling Rate> tuple, Random Forest algorithm gave the best accuracy.

In the activity recognition area, effective sampling rates and power consumption were deeply investigated issues. However, since studies that investigated the influence of window sizes on both CPU utilization and accuracy are quite few, we focused on them in this study. Moreover, to the best of our knowledge, there are no studies that investigate the accuracy, CPU utilization, sampling rate and window size together. Consequently, we believe that our study will lead future researches in this topic.

REFERENCES

[1] L. Bao and S. Intille, “Activity Recognition from User-Annotated Acceleration Data”, Lecture Notes Computer Science 3001, pp. 1-17, 2004.

[2] M.F.A Abdullah, A.F.P Negara, S. Sayeed, D. Choi and K.S Muthu, “Classification Algorithms in Human Activity Recognition using Smartphones”, International Journal of Computer and Information Engineering 6, 2012.

[3] S. Dernbach, B. Das, N.C. Krishnan, B.L Thomas and D.J. Cook, “Simple and Complex Activity Recognition Through Smart Phones”, 8th International Conference on Intelligent Environments, pp. 214-221, June 2012.

[4] T.M. Do, S.W. Loke and F. Liu, “HealthyLife: an Activity Recognition System with Smartphone using Logic-Based Stream Reasoning ”, Mobile and Ubiquitous Systems: Computing, Networking, and Services, Volume 120, 2013, pp 188-199.

[5] A. Rai, Z. Yan, D. Chakraborty, T.K. Wijaya and K. Abarer, “Mining Complex Activities in the Wild via a Single Smartphone Accelerometer”, SensorKDD ’12, August 12, 2012, Beijing, China.

[6] J.R. Kwapisz, G.M. Weiss, and S.A. Moore, “Cell Phone-Based Biometric Identification”, 2010 Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), 27-29 Sept. 2010, pp. 1-7.

[7] G.M. Weiss and J.W. Lockhart, “Identifying User Traits by Mining Smart Phone Accelerometer Data ”, *SensorKDD’11*, August 21, 2011, San Diego, CA, USA.

[8] K.Kim, M.M Hassan, S. Na and E. Huh, “Dementia Wandering Detection and Activity Recognition Algorithm Using Tri-axial Accelerometer Sensors ”, Proceedings of the 4th International Conference on Ubiquitous Information Technologies & Applications, 20-22 Dec. 2009, pp. 1-5.

[9] J.R. Kwapisz, G.M. Weiss and S.A. Moore, “Activity Recognition using Cell Phone Accelerometers ”, SensorKDD ’10, July 25, 2010, Washington, DC, USA.

[10] A.Anjum and M.U. Ilyas, “Activity Recognition Using Smartphone Sensors ”, Consumer Communications and Networking Conference (CCNC), 11-14 Jan. 2013, pp. 914-919.

[11] P.Siirtola and J.Röning, “Ready-to-Use Activity Recognition for Smartphones ” , 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 16-19 April 2013, pp. 59-64.

[12] S.Thiemjarus, “A Device-Orientation Independent Method for Activity Recognition ”, 2010 International Conference on Body Sensor Networks (BSN), 7-9 June 2010, pp. 19-23.

[13] L.Sun, D. Zhang, B. Li, B. Guo, and S. Li, “Activity Recognition on an Accelerometer Embedded Mobile Phone with Varying Positions and Orientations”, 7th International Conference on Ubiquitous Intelligence and Computing, 2010, pp. 548-562.

[14] Y.E Ustev, O.D. Incel and C. Ersoy, “User, Device and Orientation Independent Human Activity Recognition on Mobile Phones: Challenges and a Proposal ”, 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, pp. 1427-1436.

[15] S.L. Lau and K. David, “Movement Recognition using the Accelerometer in Smartphones”, Future Network and Mobile Summit, 16-18 June 2010, pp. 1-9.

[16] D.Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine ”, 4th International Conference on Ambient Assisted Living and Home Care, 2012, pp. 216-223.

[17] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra and K. Aberer, “Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach” , 16th International Symposium on Wearable Computers (ISWC), 18-22 June 2012, pp.17-24.

[18] V.Q Viet, H.M Thang and D. Choi, “Adaptive Energy-Saving Strategy for Activity Recognition on Mobile Phone”, IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 12-15 Dec. 2012, pp. 95-100.

[19] M. Jette, K. Sidney, G. Blümchen, “Metabolic Equivalents (METS) in Exercise Testing, Exercise Prescription, and Evaluation of Functional Capacity”, Clin. Cardiol. Vol. 13, August 1990

[20] M.Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD Explorations Newsletter 11, no. 1, pp. 10-18, 2009.

[21] http://web.archive.org/web/20100225065617/http://www.nytimes.com/2010/02/22/technology/22chip.html , as of August 13, 2015

[22] http://en.wikipedia.org/wiki/Apple_A4 , as of August 13, 2015

[23] http://web.archive.org/web/20100706061702/http://www.apple.com/ipad/design/ , as of August 13, 2015

[24] https://developer.apple.com/library/mac/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/Introduction/Introduction.html , as of August 13, 2015

[25] Y. M. Kawick, “Power Consumption Testing for iOS”, North Dakota State University, October 2013.

[26] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, “A smartphone-based fall detection system,” *Pervasive Mob. Comput.*, vol. 8, no. 6, pp. 883–899, 2012.

[27] G. Cola, A. Vecchio, and M. Avvenuti, “Improving the performance of fall detection systems through walk recognition,” *J. Ambient Intell. Humaniz. Comput.*, vol. 5, no. 6, pp. 843–855, 2014.