IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Enhanced User Motion Detection using Multiple Smartphone Sensors

Amit Thakor[1], Prof. Srushti P. Soni[2]

Electronics & Communication, Parul Institute of Technology, Vadodara, Gujarat, India[1,2]

Abstract: Number of research papers have been published in recent years which demonstrate how data received from sensors can be used for context aware computing. Also due the fact that nowadays smartphones come equipped with many motion sensors, people have begun developing interesting mobile applications that do automatic context aware computing. In this paper we discuss how data sensed from smartphone's motion sensors like accelerometer and gyroscope, can be used to predict user's motion type like cycling, walking, driving car and travelling in train, and furthermore do automatic motion profiling like putting the phone on vibrate mode or opening an application etc. First of all, we compared performance of various algorithms like Random Forest, J48, REPTree, Naïve Bayes, Rotation Forest (ensemble method) etc. while classifying motion types. Secondly, we pointed out the challenges faced when using only two sensors (i.e. accelerometer & gyroscope) to predict motion type and also when an identical motion type exists in data set like walking and cycling. Thirdly, we proposed a technique that makes use of an additional sensor GPS & google maps along with accelerometer & gyroscope to correct wrong predictions made by classifying algorithm to further improve the existing model with aim to deploy it for practical situation. This technique is based on usage of speed and location parameters that logically corrects wrong class. This is a promising approach when data set contains similar motion types.

Keywords: Context Awareness; Motion Type Prediction; Algorithm Comparison; Error Correction.

I. INTRODUCTION

In the last few years, context aware computing has been on boom due to evolution in electronic gadgets such as smart-phones and smart-watches. In addition to their basic functionality, these gadgets are nowadays embedded with many other sensors like accelerometer, gyroscope, compass, magnetometer, GPS, UV, SpO2, pedometer, barometeretc which were earlier available only in dedicated equipments. Moreover this System on Chip (SoC) has a benefit that allows us to simultaneously control and sense these sensors with OS such as Android or Apple iOS with ease.

Due to this advantage context aware computing has become significantly easier to implement in practical use as these gadgets are carried by people almost always. This opens a whole new dimension of exploiting possibilities from these exciting gadgets that can improve our lifestyle in terms on convenience, security or healthcare/fitness applications. In this paper we discuss how data from accelerometer & gyroscope sensor of smartphone is used to detect user's motion type and do motion profiling such as automatically enabling GPS while driving car or putting phone on vibrate mode while walking etc. The comparison of various algorithms also explores the limitation with two sensor (i.e. accelerometer and gyroscope) based model. Section II & III of this paper explains the process of data sensing from sensors and gives rough idea of the working principle. Section IV and V gives detailed analysis of result obtained and proposes solutions to the problems and limiting factors in the model.

II. DATA COLLECTION

Prior to recognition of motion type, data from accelerometer and gyroscope is recorded simultaneously at a sampling rate of 0.1 second. The data sampled from accelerometer and gyroscope is in form of x, y and z axis. In this research paper we have considered use of magnitude computed of x, y and z axis rather than analyzing each axis data individually. In our preliminary testing, while recording data for each motion type we realized that when user dynamically changed his orientation of smartphones while changing direction of walking, there were unwanted fluctuation in each axis of accelerometer and gyroscope. To counteract this problem we computed magnitude instead of data samples from individual axis which gave stable reading even while changing orientation.
We defined magnitude of accelerometer and gyroscope sensor as follows:

$$\text{Magnitude (m)} = \sqrt{(V_{x,s}^2 + V_{y,s}^2 + V_{z,s}^2)}$$

Structure of Data set Dsm is:

$$<t, V_{x,s}, V_{y,s}, V_{z,s}>$$

where 't' is time stamp of the sample and $V_{x,s}$, $V_{y,s}$, $V_{z,s}$ are the values of sensor s on the x, y and z axes, This idea is mainly inspired by research in [1] [4] and [5]. Fig 1(a) shows the accelerometer value for cycling of all three axis, while Fig 1(b) shows the magnitude value of Fig 1(a) that is cycling and Fig 1(c) shows magnitude value for driving

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

car. Even from the naked eye looking at this raw data, we can see that there is visible difference between the pattern for walking and driving. Similar differences are noticeable for different set of motion types, which allows us to discriminate between different class of motion types.

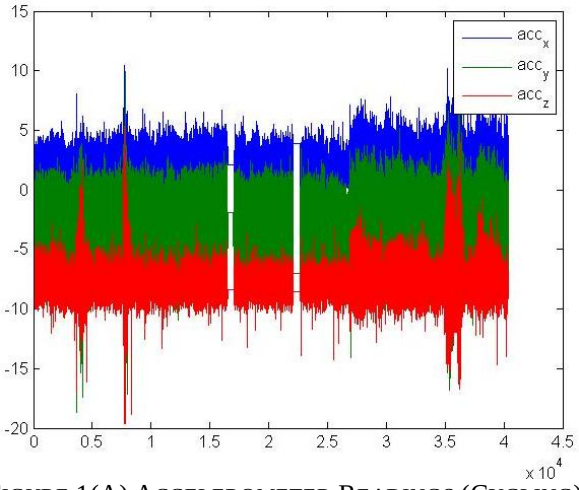

FIGURE 1(A) ACCELEROMETER READINGS (CYCLING)

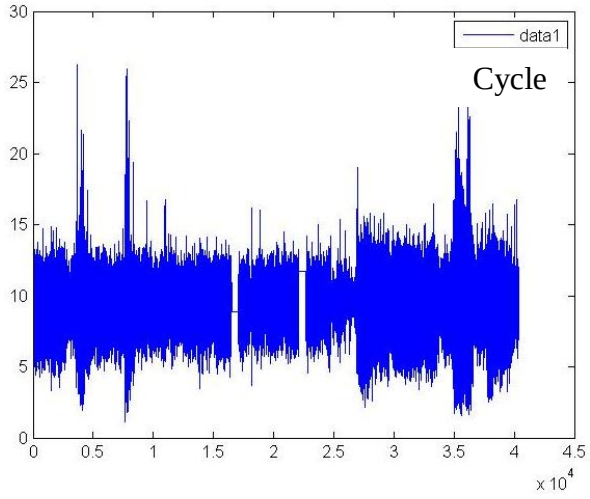

FIGURE 2(B) ACCELEROMETER MAGNITUDE (CYCLING)

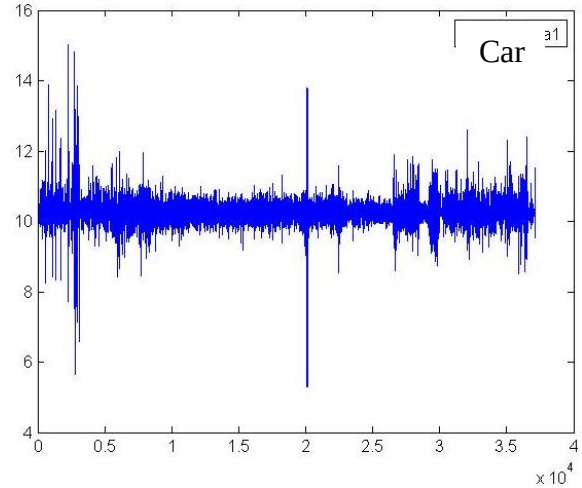

FIGURE 3(C) ACCELEROMETER MAGNITUDE (DRIVING)

III.FEATURE EXTRACTION

Post Collecting Data from sensors we extract features that discriminates between different classes of motion types. We collect four attributes {min, max, avgandstd}, that is minimum, maximum, average and standard deviation of the magnitude of the sensor under consideration. For different motion types these attributes exhibit different behavior and hence they can be discriminated. A feature set for accelerometer sensor can be given as follows:

$$F_A = \{\min(A); \max(A); \text{avg}(A); \text{std}(A)\}$$

Similarly for gyroscope sensor it can be given as follows:

$$F_G = \{\min(G); \max(G); \text{avg}(G); \text{std}(G)\}$$

By combined use of feature set from both accelerometer and gyroscope, significant improvement in accuracy of motion type is noticed. The combined feature set can be given as follows:

$$F_{AG} = F_A \mu F_G$$

IV. DATA CLASSIFICATION

Here, different classifying algorithms are used to learn the patterns, and identify the current motion type like walking, running, driving, cycling etc. In our research we considered four motion types walking, cycling, driving car and travelling in train. Our main emphasis was on 'cycling' where the motion type resembles quite a lot to walking but yet so different.

Of all the sports activity like jogging, running, swimming etc. 'cycling' is the only activity where impact collision does not exist. This is due to the fact cycling involves continuous movement of legs without any jerkiness and hence it is known as non-collision activity/sport. Inclusion of cycling activity was done on purpose as it is a popular motion type but moreover, it was the curiosity to know how Naïve Bayes algorithm performed for cycling with and without walking. In [1], although Naïve Bayes had overall low accuracy, it peaked in classifying walking motion type even bettering Random Forest algorithm which had best overall accuracy.

A motion type classifier learns the pattern and features from a training set as mentioned above in Feature Extraction section and also takes a set of features from a sequence provided and produces output in form of motion type. In research paper [1] mainly three algorithms were analyzed which were Support Vector Machines, Naïve Bayes and Random Forest. They were used to classify three motion types which were walking, driving car and travelling in train. We analyzed various other algorithms like Rotation Forest, J48, Classification Via Regression, Attribute Selected Classifier etc, over four motion types including cycling, driving in car and travelling in train. For evaluation of different algorithms we used a machine learning software **"WEKA"**. The data we collected was formatted into '.ARFF' file which is acronym for Attribute Relation File Format' as WEKA only accepts .ARFF

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

format. As we discussed about differences in extracted feature for different motion types, based upon it WEKA classifies motion types using the algorithm under test. In our testing we recorded data samples for walking, cycling, driving car and travelling in train, each activity with at least over 1 hour recording.

A. Comparison of Algorithms

In CASE 1 we created data set comprising all four motion types(i.e. walking, cycling, driving and travelling in train) and predicted the motion typeby using various algorithms. In CASE 2 and CASE 3 we did classification amongst only three motion types, swapping walking and cycling in each case, the former being simulation of situation as in [1].

TABLE I (CASE 1) COMPARISON OF ALGORITHM FOR ALL FOUR MOTION TYPES

	Random Forest	**Naïve Bayes**	**J48**	**REPTree**	**Classification Via Regression**	**Rotation Forest**
Walk	70.50 %	60.90 %	72.60 %	73.60 %	73.50 %	72.50 %
Cycle	74.90 %	82.80 %	80.10 %	78.30 %	80.10 %	80.30 %
Car	90.80 %	87.90 %	93.90 %	93.40 %	93.40 %	93.40 %
Train	84.00 %	86.10 %	82.90 %	83.20 %	83.20 %	89.70 %
Overall	79.11 %	77.98 %	81.82 %	81.54 %	82.05 %	81.86 %

TABLE II (CASE 2) ACCURACY OF DIFFERENT ALGORITHMS FOR THREE MOTION TYPES EXCLUDING CYCLING

	Random Forest	**Naïve Bayes**	**J48**	**REPTree**	**Classification Via Regression**	**Rotation Forest**
Walk	99.40 %	99.70 %	99.60 %	99.60 %	99.60 %	99.69 %
Car	92.10 %	88.40 %	94.60 %	94.60 %	95.00 %	94.70 %
Train	84.00 %	86.50 %	83.50 %	83.30 %	83.30 %	83.50 %
Overall	93.50 %	92.75 %	94.33 %	94.24 %	94.38 %	94.39 %

TABLE III (CASE 3) ACCURACY OF DIFFERENT ALGORITHMS FOR THREE MOTION TYPES EXCLUDING WALKING

	Random Forest	**Naïve Bayes**	**J48**	**REPTree**	**Classification Via Regression**	**Rotation Forest**
Cycle	97.70 %	97.10 %	98.20 %	98.20 %	98.20 %	98.10 %
Car	91.00 %	87.90 %	93.60 %	93.40 %	93.50 %	93.50 %
Train	84.20 %	86.20 %	83.00 %	83.20 %	83.30 %	83.60 %
Overall	92.17 %	91.25 %	93.11 %	93.05 %	93.14 %	93.17 %

In CASE 1 where all four motion types are considered, we can notice that the overall accuracy of all algorithms considered was quite low in comparison to CASE 2 & 3 where cycling and walking were excluded. The confusion matrix suggested that most of the errors in classification occurred in CASE 1 was due to confusion between walking and cycling which had a bit similar motion pattern. As the number of motion types were increased, the overall accuracy decreased. Of all the algorithms considered, classifier ensemble method 'Rotation Forest'performed best in almost all cases with different set of motion types as in CASE 1,2,3 or other sets comprising only cycle and car etc. The most notable point about Naïve Bayes was that it performed well in classifying walking and cycling as in CASE 2 and CASE 3, but it took a steep fall when both walking and cycling were paired together as in CASE 1. Naïve Bayes ended up piling walking instances as cycling which shows its highly unreliable nature when several motion types are put together especially identical ones. Although all algorithms were affected in situations like this but Naïve Bayes was most vulnerable of all. To counter-act this problem we proposed a technique which makes use of an additional sensor like GPS and Google Maps along with accelerometer and gyroscope.

B. GPS & Google Maps

GPS (Global Positioning System) and Google Maps provide real time information about user's current speed and location in terms of latitude, longitude and altitude.The location accuracy may slightly deviate from the real location by ± 30-35 meters at max, which is good enough to fulfil our goal. The 'speed' and 'location' parameters are additional discriminating factors that may help us logically classify motion types. Combining this quality with the

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

result obtained from accelerometer and gyroscope we can greatly decrease the classification errors and increase accuracy with ease. The next section elaborates our proposed method and its working, and how its output may be used for motion profiling.

V. ERROR CORRECTION LOGIC

As mentioned in the above sections about the problem in classification of motion types with identical pattern, we demonstrate here how the predicted classes by algorithms can be corrected by applying simple logic which uses speed and location parameters. This is basically post processing of the result obtained from algorithms which is then used to do motion profiling. Figure 2(A) is the graphical representation of our Error Correction Logic.

The column in the red box on left hand side is the output of classifier and the column right next to it is the actual class of instance in the first and second column. The logic is as follows: For example, if the predicted class is walking and the speed at that instance is for example greater than 6 km/h then the class is corrected to cycling as no average person would walk at over 6 km/h speed. Similarly if the speed is less than 6 km/h and prediction is 'cycling', it can be corrected to 'walking'. Another logic to correct class may include location parameter where location is a discriminating factor like in case of train and car. For example, if predicted class is 'car' but Google Maps' API suggests that the location at that instance was in the vicinity of railway tracks, the class can be corrected to 'train'.

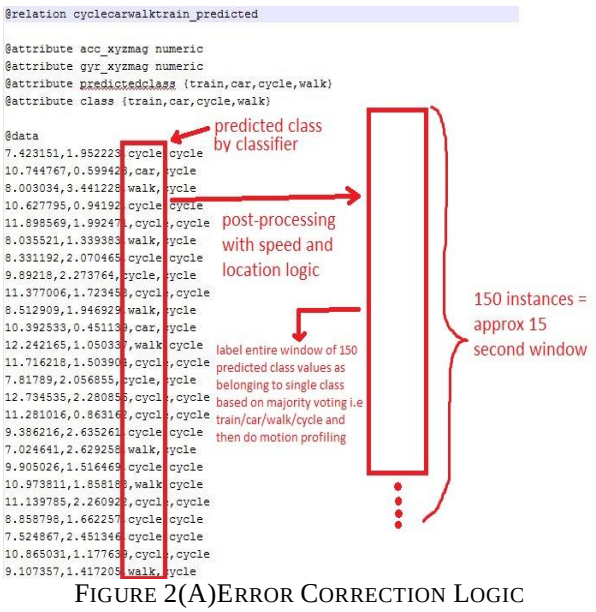

FIGURE 2(A)ERROR CORRECTION LOGIC

The new column in red box on right hand side is the correct result after post processing with the above logic. We can segment this column into small windows of 150 classes (which is equivalent to 15 second window) and label this window as a single class by majority voting. This is done since no user will change his motion type so rapidly within a span of 15 seconds. After labelling this 15 seconds window with a class we can instruct the android application to do motion profiling or take particular action based on the class.

To test the performance of our Error Correction Logic, we recorded a fresh set of data called 'test set' of 40 minutes. It comprised of all four motion types, 10 minutes each. We used Rotation Forest algorithm's model (generated from our training set) to do prediction on the 40 minutes test set. It achieved 76.05% accuracy, correctly classifying 18199 instances out of 23928 instances.

Figure 2(B) shows the bar graph of predicted class without ECL. As discussed earlier, the stats showed high number of inter-classification errors between 'walking' and 'cycling'. The bar graph on right shows the number of instances with their classes and the ones on right side shows the number of instances with their predicted classes. Bar graph suggests how similar motion types like 'car' and 'train' suffered a bit from inter-classification errors but they were much less in comparison to 'walking' and 'cycling'. So we post processed the output with our error correction logic using only 'speed' parameter to correct the errors related to 'walking' and 'cycling'.

The outcome was impressive as the inter-classification errors were drastically reduced and the overall accuracy shot up from 76.05% to 90.77%. This can be seen in Figure 2(C) which shows the result after application of our Error Correction Logic.

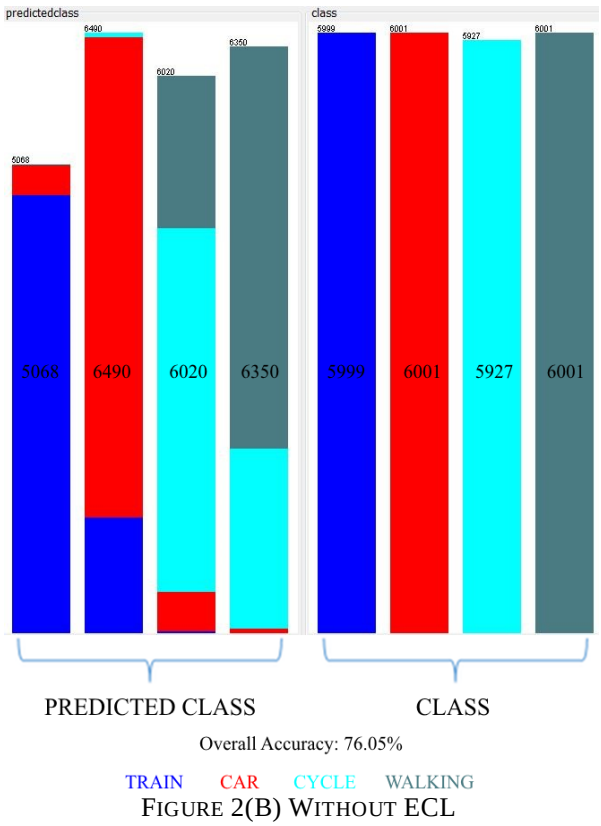

FIGURE 2(B) WITHOUT ECL

DOI10.17148/IJARCCE.2017.63235

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

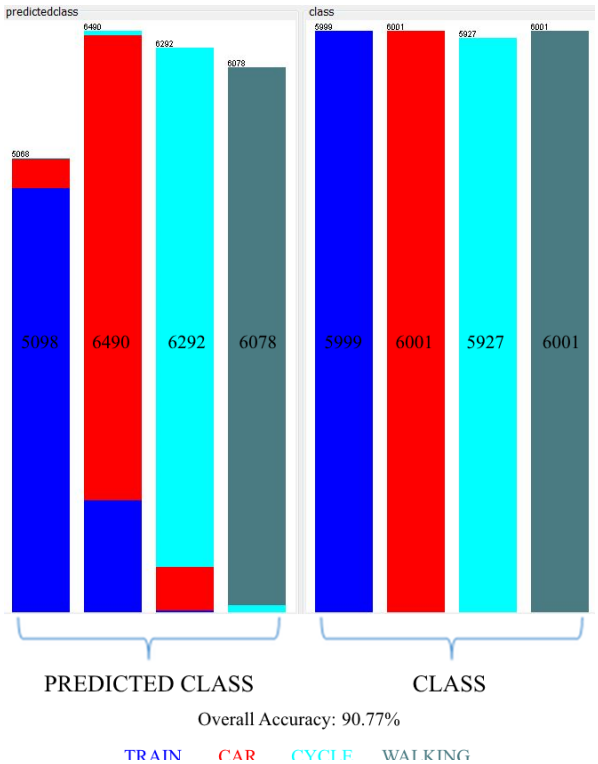

FIGURE 2(C) WITH ECL

Although the inter-classification errors between 'car' and 'train' were untouched as they depended on location parameter, correction of 'walking' and 'cycling' alone made a huge difference. Furthermore, we felt that correctingerrors of 'car' and 'train' based on 'location' parameter may increase accuracy or might make the matter worse depending on the accuracy of GPS. Although there is a risk in closed looped system, where a system might become more unstable in attempt to over correct the deviation from desired outcome, here the tradeoff was far less and the decision to correct 'walking' and 'cycling' using 'speed' parameter really paid off.

VI. SENSE ME ANDROID APPLICATION

To implement our system on an android smartphone, we developed and android application called 'Sense Me' that is able to detect user's motion type in real time or do prediction of an ARFF file provided by the user, using our generated model from our training set. The application is equipped with four of our algorithm models, namely Rotation Forest Ensemble Method, Random Forest, J48, and Naïve Bayes. Although at this stage we have only utilized 'speed' parameter, user can choose to do post processing of output by our Error Correction Logic (ECL) to further increase accuracy. Figure 3(A) shows the graphical user interface of 'Sense Me' Application. The application interfaces with Google Android's API to sense the readings from sensors and predict user's motion type.

The application can run in the background and continuously record readings from sensors, while the user is free to use the smartphone however he likes. These kind of context aware applications can be used to automate tasks or simply analyze user's daily activities by keeping a log.

FIGURE 3(A)SENSE ME ANDROID APPLICATION

VII. CONCLUSION

In this paper, we discussed how above four motion types can be classified with the help of accelerometer and gyroscope of a smartphone, and also compared performance of various algorithms. Most importantly we highlighted the limitation of dependence on only accelerometer and gyroscope. To counteract the above, we proposed a tested a technique that involves use of an additional sensor (i.e. GPS & Google Maps), to reduce classification errors and significantly improves the model's accuracy. Future Work Plan: Finalization of our android application for implementation on Android based smartphone and launch on Google Play Store.

ACKNOWLEDGEMENT

We would humbly like to extend our gratitude to **Mr. Luca Bedogni,** Department of Computer Science, University of Bologna, for his continuous support and guidance.

REFERENCES

[1] By Train or By Car? Detecting the User's Motion Type through Smartphone Sensors Data. IEEE, Wireless Days (WD), 2012 IFIP DOI 10.1109/WD.2012.6402818.

[2] Towards Physical Activity Recognition Using Smartphone Sensors. Shoaib, M., Dept. of Comput. Sci. &Electr. Eng., Univ. of Twente, Enschede, Netherlands, Scholten, H. ; Havinga, P.J.M. Vol 978-1-4799-2481-3.

[3] Formulation of a simple model to estimate road surface roughness condition from Android smartphone sensors. Intelligent sensors, sensor networks and information processing (ISSNIP), 2014 IEEE Ninth International conference on DOI: 10.1109/ISSNIP.2014.6827694.

[4] A smartphone localization algorithm using RSSI and intertial Sensor Measurement Fusion. William Wei Liang Li, member IEEE, Ronalg A Eltis, Senior member IEEE, and Moe Z Win, Fellow IEEE. Vol 978-1-4799-1353-4, 2013 IEEE.

[5] Activity Recognition with Smartphone Sensors. Published in Consumer Communications and Networking Conference (CCNC), 2013 IEEE. DOI: 10.1109/CCNC.2013.6488584. Page No 914 – 919.

[6] Ravi, N., Dandekar, N., Mysore, P., Littman, M. L. (2005). Activity Recognition from Accelerometer Data. Neural Networks: 20(3), pp. 1541-1546, 2005.

[7] http://developer.android.com/guide/topics/sensors/sensors_motion.html

[8] Rodriguez J.J, Kunchev, L.I., Alonso, C.J., Rotation Forest: A New Classifier Ensemble method. Published in 'Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2006.211, page(s) 1619-1630.

[9] Breiman, L. Random forest. Machine Learning, 45(1), pp. 5-32, 2001

[10] C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers. ISBN 1-55860-238-0.

[11] Mobile Sensor Data Collector using Android Smartphone. Published in "Circuits and Systems (MWSCAS)", 2012 IEEE 55th International Midwest Symposium. DOI: 10.1109/MWSCAS.2012.6292180. Page No 956 – 959.

[12] Ultra Low Cost Radiation monitoring system utilizing smartphone connected sensors developed with internet community. Yang Ishigaki, Yoshinori Matsumoto, Ryo Ichimiya and Kenji Tanaka (University of Tokyo. Vol 978-1-4577-1767-3, 2012 IEEE.