

# Development of Motion Detection Algorithms Based on Simultaneous Execution Using Mobile Phone Sensors

Zsófia SÁNDOR<sup>1</sup>, Gergely KIS<sup>2</sup>

<sup>1</sup>Data scientist, research manager, Budapest, e-mail: sandorzsofi@yahoo.com

<sup>2</sup>Department of Infocommunication, Corvinus University of Budapest, Budapest,  
e-mail: gergely.kis@uni-corvinus.hu

Manuscript received February 15, 2016; revised December 24, 2016

**Abstract:** The proliferation of sensor networks employing wireless data transmission technologies has paved the way for the collection of large amounts of measurement data. Several research teams have used this opportunity to develop algorithms aimed at gaining information from sensor data. Motion detection is one of the most actively researched areas. In this article, we present a system for examining motion detection in a general environment. In other words, motion forms are not identified with various wearable sensors; instead, we use the data collected by the sensors of mobile phones kept with almost all members of society now.

**Keywords:** Motion detection, sensor data.

## 1. Introduction

A boom of wireless communication technologies and specifically mobile phones has brought about institutional, social and cultural changes, just as predicted by the American writer George Gilder [1]. Technological innovation offers solutions to an increasing number of issues that used to seem insoluble due to technological constraints and the lack of measured data.

Within technological innovation in general, progress in terms of decreasing calculation costs has been especially fast. Consequently, tasks can now be tackled that require the processing of large quantities of information. A current buzzword is “Big Data”, meaning the handling of data quantities that cannot be managed with conservative data processing methods and tools. Various sensory information is a typical source of such high-volume data. The second relevant direction of technological development is related to sensor development: sensors are decreasing in size, becoming more and more accurate, and consume less energy. The spread of wireless sensor networks (WSN) allows for the real-

time transfer and processing of the data collected by a sensor, and also for connecting several sensors.

Studying human behavior is a key social research area. Gaining an insight into human behavior could yield widely applicable knowledge, for example in the fields of medicine, psychology, economic marketing, and health care. One of the projections of human behavior is physical movement; observing movement may lead to conclusions about the individual's behavior and habits.

Thus, scientists are facing a specific range of issues, but also have the toolset that could provide answers to those issues. Hence today's wide-ranging research is made about the usability of sensor-collected data for motion detection. Most of that research is conducted in laboratories, with sensors attached to test subjects' bodies.

This article presents a system based on the processing of data generated in everyday life, outside of the laboratory environment. The definition of the problem at hand is followed by the description of the data collection and transfer system, by the presentation of processing algorithms built on each other, and finally by a summary and conclusions.

## **2. Problem definition**

### *A. Historical research*

Sensors built into smartphones and other wearable devices can be used in a variety of ways. With the right method, they are suitable for detecting practically all forms of motion, from basic step recognition to recognizing complex actions and providing motivation for a healthy lifestyle. This variety (both in terms of methods and areas of utilization) is reflected by international technical literature.

Some research is aimed at specific areas, e.g. at finding the most accurate step counting method [2]. Using accelerometers, the margin of error is just a few steps. In an advanced version of this solution, speed can also be taken into consideration as a factor [3]. In this research, steps were counted during slow and fast walking, downhill and uphill walking, and while climbing stairs. The researchers used gyroscopes because they had found that an accelerometer does not yield accurate step numbers if the test subject walks slowly. The results are encouraging: the accuracy of step counting in slow walk was above 96% on a flat surface, more than 95% on inclining and declining routes, and higher than 90% when climbing stairs.

Basic actions such as walking, running, climbing stairs, sitting, standing, using an elevator, and jumping have been examined by several research teams. Some of these achieved 99.97% accuracy in action recognition using the

Random Forest method [4], while the Nearest Neighbor method yielded 93% accuracy in [5]. The limitations were only one phone, held in the test subject's trouser pocket in [4]; in another case [5], a single phone model was used, but at two locations.

Resting positions (standing, sitting, lying) can be detected highly accurately [6]. Besides these, walking and climbing stairs were examined in this research, which resulted in 86% accuracy using a decision tree. Others went farther than that, detecting car driving in addition to walking, running, cycling, sitting and standing in [7]. In the research, the effectiveness of the QDA (Quadratic Discriminant Analysis) and the k-Nearest Neighbor algorithms was tested in online and offline mode. In online mode, the accuracy of QDA was 95.8%, and that of k-NN was 93.9%; offline, QDA yielded 95.4%, and k-NN 94.5%. The phone position was fixed in these cases, too, and in research no. [7], its direction was also fixed.

In [8], the goal was to identify actions regardless of the position of the phones; but in this case, too, only the basic actions were examined.

In certain publications [9, 10, 11], the objective was to present a connection between a healthy lifestyle and movement detection. In one case [9], the researchers developed a step counter using accelerometer data and a neural network, aiming to detect false steps (a common error in step counters). The ultimate goal was to increase the reliability of the health preservation system. In another research, algorithms were developed for the recognition of uphill and downhill walking, walking on a flat track, climbing and descending stairs, and running in [10]. Again, the goal was health preservation. The accuracy of recognizing these actions was 93.2%, 97.4%, 97.6%, 98.8%, 92.2%, and 90.8% respectively. Not all analyses are aimed at the processing of health data: based on a market approach, a convenience function has been developed, focused on automatically changing the phone's settings at the start of certain activities such as running [11].

Besides the basic actions, the recognition of complex activities (such as hand washing, house cleaning, cooking etc.) has also been researched in [12]. A wide range of methods was employed. These (apart from the Naïve Bayes method) were suitable to identify basic actions with over 90% accuracy. However, the highest accuracy in the case of complex actions was merely 50%.

In most cases, wearable sensors provide good estimations for the user's activities; because of the fixed location [13, 14, 15].

As shown above, research is progressing towards several dimensions of complexity. Our goal is to separate as many medium-complexity actions as possible. (These include, for example, riding a bus, but not dish washing.) For wide-ranging usability, that should be achieved with mobile phones regardless of model and location.

### *B. The challenge*

The objective of the system presented is to accurately identify movement in its natural environment, without additional devices. A solution may not become wide-spread if the users find the required device inconvenient or uncomfortable. One example is the need to use an ankle strap and a wristband at the same time may force users to change their daily routines. This is why mobile phone sensors were selected in our solution. People keep their phones with them during the active part of the day, so the phones usually move together with the users.

However, mobile phones present challenges other than algorithm development.

The first challenge is related to eliminating the above-mentioned inconvenience. Mobile phone usage is convenient if the system meets certain basic requirements such as low energy usage and data transfers, and optimized resource (e.g. memory) usage. These requirements must definitely be met for a solution to be easy to use.

The second group of challenges arises from variances across both users and devices. Movement detection is difficult because of differences between people's physiques, habits and movements; the same actions vary from person to person if we rely on time-series data of sensors. In addition, differences between mobile phone makes and models should also be addressed; the quality and accuracy of built-in sensors vary, and even the measurement units may differ. These challenges need to be resolved for the wide-spread proliferation of a system.

The third challenge involves ensuring the right response time with sufficient accuracy. The system will not be used in a lab environment, so, according to international literature, we should not expect the models to be as accurate as in a lab, especially because some of the activities to be differentiated are similar to each other.

### *C. Technical solution concept*

Before outlining the solution concept, it is important to note that two options are currently available to researchers concerning the technicalities of data transmission and processing. Each of these options has its advantages and disadvantages, as explained below.

The problem is that sensor data are available in the mobile phone or other sensor-equipped device. They either need to be transferred to a processing server, or the processing must be performed by the device itself. In the former case, the issue of high data transfers – and thus higher energy consumption – must be resolved as this may be inconvenient to users if the data are transferred

via a mobile Internet connection. Also, the increased data intensity means that the server's capacity must be sufficient. On the other hand, the advantage of this option is that there is no data loss.

The main argument for in-device processing is that not all the data are required; algorithms running on the device can be used to obtain material information, and only that is transferred to a server. This is a major advantage because it saves both device and server resources. On the other hand, running the processing algorithms uses more memory, and the running jobs make the energy consumption higher, which can also be inconvenient to users. The advantages and disadvantages of the two methods are summarized in *Table 1* below.

*Table 1:* Assessment of data processing methods

	<b>Data processing on server</b>	<b>Data processing on device</b>
<b>Amount of data to be transmitted</b>	high	low
<b>Local memory usage</b>	low	high
<b>Server storage space requirement</b>	high	low
<b>Information loss</b>	no	yes
<b>Energy cost</b>	higher due to the data transfer	higher due to the running algorithms

We have opted for data processing by a server, primarily because we strove to keep all information obtained during the research.

In order to resolve the above issues and tackle the challenges, the system shown in *Fig.1* has been created. As we collect data from mobile phone sensors, a mobile phone application is one of the central elements of the system. This application collects data based on the configuration received from the messaging server, writes those data into a file, compresses the file, and transfers it to the processing servers in the required format. The role of the data server is to pre-process the high quantity of received data, and to send them to the algorithm servers. The processing servers run the algorithms, and then estimate which activities the user is probably performing. The final decision is made based on several partial decisions.

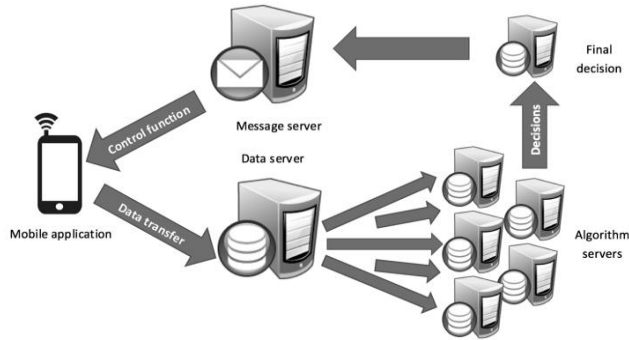


Figure 1: Data processing

### 3. Measurement and data collection

Defining the right data collection methodology is essential both for modelling and maintaining the operability of the system. As customizability as well as easy and convenient usage are paramount to users, we aimed not only for collecting and transferring data in the highest quality possible, but also for optimal resource utilization.

The mobile phone application we have developed has a built-in algorithm to sense low-activity periods when data are not collected. Naturally, this so-called sleeping mode is customizable via the configuration of the messaging server, i.e. the running and discontinuation of measurement can be controlled. The sleep mode optimizes both data traffic and battery usage and also the energy consumption.

Sensor data are obtained through the sensor's API (Application Programming Interface) and are stored in a manner that minimizes data size as much as possible. This size reduction consists of two parts: transforming and encoding the file in an optimized size; and compression. The resulting file (ready for transfer) is about 12% of the original data size. In addition to this size reduction, sleep mode may significantly lower the data transfer volume, depending on the activity ratio. The range of sensors to be used for measurement can also be configured; the algorithms have been developed with the accelerometer, GPS and gyroscope in focus.

Transferring the collected data to the server in the manner described above causes minimal inconvenience to the user. This means that the energy consumption is reduced, and also the amount of processing on the device itself

keeps the user experience high. All further work phases of the system are independent of the user, i.e. they do not influence the user at all.

The data are received by the data server's pre-processing module which prepares the data for processing, i.e. decompresses and decodes them, and runs processes required for subsequent algorithms. These include ensuring the coherence of the time series, decreasing their dimensions, and interpolation in order to eliminate inaccuracies (minor differences in sampling at a given frequency) in the sensor API. One of the gravest problems of data collection with mobile phones is the variance in the type and quality of sensors built into the devices of various manufacturers; the measurement scales may differ, and the accuracy of the sensors almost certainly varies. Figure 2 shows deviations in accuracy. The y axis is the acceleration in  $\text{m/s}^2$ , and the x axis is the millisecond of the measurement.

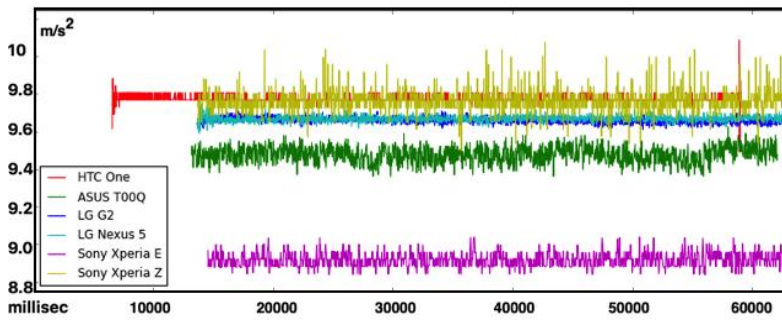


Figure 2: Comparison of the operation of mobile phone sensors

Fig. 2 shows time series of the accelerometer sensors of the mobile phones tested. All phones were stationary, but the sensor readouts indicate different mean values and standard deviations. These differences must be managed with algorithms. The figure above shows data from devices running the Android operating system only; iOS measurements are very different due to the data being recorded in different measurement units. Android phones measure acceleration in  $\text{m/s}^2$ , while iOS in  $g$ . The gravity in common is measured in units of acceleration,  $g$  means the measurement unit of gravity, thus  $1 g$  equals to approximately  $9.81 \text{ m/s}^2$  in our case. So unifying the measurement scales is an important task within pre-processing phase.

## 4. Algorithm development

### A. Activities

The first step in algorithm development involves defining the activities to be identified. Several scenarios are examined in the related technical literature

using the methods described above, from simple step count to attempting to identify the placement of the device. In our own research, we attempted to identify actions regardless of the placement of the user's phone, i.e. we seek patterns that are common to all phone placements. As an example, let us take a phone kept in the user's trouser pocket or purse. These cases differ because a phone kept in a trouser pocket moves together with the user's body, while a telephone kept in a purse moves partly independently of the body. But the basic movement pattern must be the same in both cases; it is this pattern that we aim to find with algorithms, ignoring the noise caused by the placement of the device.

During the research, we collected as detailed reference data as possible, but some of the actions were aggregated because they were similar. Reference measurements for the following actions were made:

- Human movement: immobility (phone placed on a desk or kept with the user); walk; run; ascending and descending stairs; phone usage (calls and screen usage);
- Vehicular movement: bus; metro; tram; trolley; train; suburban train;
- Other: lift and escalator (ascending or descending).

### *B. Concurrent algorithms*

After pre-processing of accelerometer data, the system processes branch off so that several algorithms can run concurrently for the best possible end-result. As each algorithm has strengths and weaknesses, they would not be sufficient individually. But they support and improve each other for a robust overall system.

For example, one algorithm, which is based on GPS usage, can significantly improve the results of other algorithms. But for that, the user must turn on GPS-based location identification in the phone's menu; and even if that is enabled, there is no GPS connection indoors. So, that algorithm is used as an accessory function only; our algorithms are mostly based on an accelerometer.

Pre-processing makes the data easily usable for the concurrent algorithms. This means, that all the data packages that the algorithm servers receive do not need any additional calculations.

There are two types of algorithms. The first type is when we identify episodic types of events while the second is when we identify the activities as macro-processes.

All of the activities have typical episodes. For example, when a vehicle accelerates; when we take a step; when we sit down and so on. These episodes last for very short periods but they are pretty easy to identify. The first algorithm catches these episodes and is based on clustering. The variables of the clustering are the characteristics of the time series. We used the k-means

clustering algorithm for generating the static clusters for the episodes, and the distances from the cluster centers helps us to assign the new samples to the clusters and recognize the activities this way.

The second group is the macro-process type algorithms. The first amongst these algorithms is the GPS algorithm. The GPS algorithm is mostly based on the calculation of the speed based on the time and the distance. GPS coordinates are pretty accurate, so we can use the haversine formula to calculate the distance between two coordinate pairs (1).

$$d = 2r * \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos \phi_1 * \cos \phi_2 * \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Where  $r$  is the radius of the Earth that is approximately 6371km,  $\lambda_2 - \lambda_1$  is the difference of the two longitudes and  $\phi_2 - \phi_1$  is the difference of the two latitudes. The computed and corrected speed values help us to identify the activities, as the activities can be well separated using these values.

The second algorithm of this group is based on the similarity of the 3-dimension accelerometer time series. Time series data points can be paired to each other and – as acceleration is measured in 3-dimensions – Euclidean distance can be calculated in 3 dimensions (2). Using the distances, activities can be categorized by comparing the differences to the reference data.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \quad (2)$$

The third algorithm of this group is based on the amplitude of the time series. This algorithm is the best in recognizing the real macro activities. This is able to recognize the stops of a public transport vehicle and this way it can identify the travelling on public transportation as a process. The highest and lowest amplitude sections' patterns are unique by the activities.

The last algorithm in this group is based on a matrix model. In this model, we create unique “masks” of each reference activities. Activities differ in the sense how data points follow each other so if we use a matrix to identify the sequentiality of the data points, we can create matrices that can be used to identify the unknown activities. These “masks” create a similarity measure that varies between -1 and 1, and where 1 means the identity.

### *C. Real life examples*

Table 2 and 3 each show the similarities/differences between the activities for one algorithm. The figures indicate the similarity of the selected basic actions.

*Table 2: Similarities of basic actions as indicated by sensor data*

	bus	stairs	lift	metro	immobility	walking	phone usage	tram	train
bus	0,91	0,67	0,25	0,70	0,31	0,66	0,71	0,54	0,65
stairs	0,67	0,95	0,36	0,46	0,51	0,95	0,43	0,39	0,46
lift	0,25	0,36	0,89	0,45	0,77	0,40	0,06	0,62	0,43
metro	0,70	0,46	0,45	0,77	0,44	0,47	0,58	0,66	0,70
immobility	0,31	0,51	0,77	0,44	0,82	0,55	0,12	0,56	0,43
walking	0,66	0,95	0,40	0,47	0,55	0,97	0,43	0,41	0,48
phone usage	0,71	0,43	0,06	0,58	0,12	0,43	0,80	0,38	0,59
tram	0,54	0,39	0,62	0,66	0,56	0,41	0,38	0,70	0,63
train	0,65	0,46	0,43	0,70	0,43	0,48	0,59	0,63	0,82

*Table 2* contains the similarity measures of the matrices that we have created in the matrix model. The theoretical minimum value is -1, while the theoretical maximum is 1. The main diagonal indicates the average similarity between identical actions, while the rest shows the distance between the actions. Note the strong similarity between walking and stair climbing, which is caused by the fact that stair climbing consists of steps, too. Interestingly, the differences between users' movements are so big that the two actions can hardly be distinguished. The only difference between them is in their amplitude, which this algorithm is less suitable to detect. Note the similarity of moving in an elevator and being immobile; the reason is that a lift usually moves in a straight line at a steady pace apart from the initial acceleration and the concluding deceleration, i.e. an accelerometer indicates the same data as in the case of immobility. Acceleration is zero both in case of immobility and during straight movement at a steady speed.

*Table 3* shows Euclidean distances in the 3-dimension accelerometer time series measured in  $\text{m/s}^2$  between reference activities – hence the zeros in the main diagonal. Some activities constitute groups of stronger similarity; and walking is similar to climbing stairs in this case as well. Also, the data measured in vehicles are similar. But this algorithm seems to better distinguish riding an elevator from a state of immobility, so it is important to combine several algorithms.

Table 3: Distances between reference actions

	immobility	walking	stairs (down)	stairs (up)	bus	metro	tram	lift	escalator	train
immobility	0,00	4,42	4,22	4,49	0,39	0,49	0,60	0,38	0,25	0,22
walking	4,42	0,00	2,46	2,80	4,10	4,02	4,43	4,18	4,28	4,23
stairs (down)	4,22	2,46	0,00	2,50	3,94	3,82	3,88	3,97	4,07	4,05
stairs (up)	4,49	2,80	2,50	0,00	4,20	4,10	4,40	4,28	4,63	4,32
bus	0,39	4,10	3,94	4,20	0,00	0,27	0,35	0,33	0,24	0,24
metro	0,49	4,02	3,82	4,10	0,27	0,00	0,38	0,32	0,33	0,32
tram	0,60	4,43	3,88	4,40	0,35	0,38	0,00	0,47	0,45	0,44
lift	0,38	4,18	3,97	4,28	0,33	0,32	0,47	0,00	0,30	0,30
escalator	0,25	4,28	4,07	4,63	0,24	0,33	0,45	0,30	0,00	0,15
train	0,22	4,23	4,05	4,32	0,24	0,32	0,44	0,30	0,15	0,00

For robustness, development was supported by large-sample data collection in order to map differences between users and to avoid “over teaching”, a frequent error when developing models. In this modeling case, it would be rather problematic if the algorithm learned too much from a single user’s movement and used that to draw conclusions about the movement of other people who walk slower or use a different means of public transport the patterns of which cannot be observed in the first user’s case, so the algorithm could not recognize them. We involved 80-100 users (in two phases) to help test the system under everyday circumstances.

The system can provide the final guesses concerning the activities performed in 2-4 minutes from receiving the data files, and is scalable depending on the number of users. Based on our back-testing (consisting of a 90-minute combination of city travel and actions by a small sample of users), the accuracy of the guesses was 67%.

## 5. Conclusion

The system presented in this article is suitable for detecting motion in its most natural form, with devices that are the most widely available. The conclusion of the research is that it is worthwhile to define several algorithms that complement each other, in order to improve the accuracy of action detection.

It is hard to compare the results to the results of other researches in the literature, because the goal of the research was pretty different from the previous solutions. In this paper our goal was to recognize the motions in their

natural way, and not in a laboratory environment. We have used a wide range of devices, more people, and several device positions. We also tried to recognize complex activities (such as traveling on tram). The statistical performance of this system is naturally lower than in a laboratory environment, and also compared to simpler models aimed at identifying just a few actions. As we could also see in the literature, accuracy decreases as we allow more and more flexibility in the system; however, this system, too, becomes more accurate as actions are aggregated or flexibility is reduced. The closest research found in the literature is [12], and compared to the 50% accuracy, our 67% is satisfactory. The results of this research provide satisfactory answers to the challenges explained at the beginning of the article. A balance has been created between the need for accuracy and the objective of universality. A system has been worked out that can be used conveniently by anyone and offers sufficiently fast response times. Further fine-tuning is underway.

## References

- [1] Freeman, C., Louca, F., “The Emergence of a New Techno-Economic Paradigm: The Age of Information and Communication Technology (ICT)”, *As Time Goes By – From the Industrial Revolutions to the Information Revolution*, New York, 2002, pp. 318–324.
- [2] Naqvi, N. Z., Kumar, A., Chauhan, A., Sahni, K., “Step Counting Using Smartphone-Based Accelerometer”, *International Journal on Computer Science and Engineering*, 4(3), pp. 675–681, May. 2012.
- [3] Jayalath, S., Abhayasinghe, N., Murray, I., “A Gyroscope Based Accurate Pedometer Algorithm”, Presented at *International Conference on Indoor Positioning and Indoor Navigation*, Oct. 2013.
- [4] Cruz-Silva, N., Mendes-Moreira, J., Menezes, P., “Features Selection for Human Activity Recognition with iPhone Inertial Sensors”, Presented at *Portuguese Conference on Artificial Intelligence*, 2013.
- [5] Das, S., Green, L., Perez, B., Murphy, M., “Detecting User Activities using the Accelerometer on Android Smartphones”, Jul. 2010.
- [6] Cerqueira da Silva, J. R., “Smartphone Based Human Activity Prediction”, *M.S. thesis, Faculdade de Engenharia, Universidade do Porto.*, 2013.
- [7] Siirtola, P., Röning, J., “Recognizing Human Activities Userindependently on Smartphones Based on Accelerometer Data”, *International Journal of Interactive Multimedia and Artificial Intelligence*, 1(5), pp. 38–45, Jun 2012.
- [8] Khan, A. M., Lee, Y.-K., Lee, S. Y., Kim, T.-S., “Human Activity Recognition via An Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis”. Presented at *Future Information Technology, 2010 5th International Conference*, May. 2010.
- [9] Tomlein, M., Bielik, P., Krátky, P., Mitřík, S., Barla, M., Bieliková, M., “Advanced Pedometer for Smartphone-based Activity Tracking”, Presented at *Proceedings of the International Conference on Health Informatics*, 2012.
- [10] Shin, J., Shin, D., Shin, D., Her, S., Kim, S., Lee, M., “Human Movement Detection Algorithm Using 3-Axis Accelerometer Sensor Based on Low-Power Management Scheme for Mobile Health Care System”, Presented at *GPC'10 Proceedings of the 5th international conference on Advances in Grid and Pervasive Computing*, May. 2010.

- 
- [11] Kwapisz, J. R., Weiss, G. M., Moore, S. A., “Activity Recognition using Cell Phone Accelerometers”, *ACM SIGKDD Explorations Newsletter*, Dec. 2010.
  - [12] Dernbach, S., Das, B., Krishnan, N. C., “Simple and Complex Activity Recognition Through Smart Phones”, Presented at *Intelligent Environments (IE), 2012 8th International Conference*, Jun. 2012.
  - [13] Yang, C.C., Hsu, Y.L., “Development of a wearable motion detector for telemonitoring and real-time identification of physical activity”, *Telemed. J. E. Health*, no. 15, 2009, pp. 62–72.
  - [14] Karantonis, D.M., Narayanan, M.R., Mathie, M., Lovell, N.H., Celler, B.G., “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring”, *IEEE. Trans. Inf. Technol. Biomed.* no. 10, 2006, pp. 156–167.
  - [15] Ozdemir, A. T. and Barshan, B., “Detecting Falls with Wearable Sensors Using Machine Learning Techniques” *Sensors*, vol. 14, MDPI, pp. 10691–10708, 2014.