



Dynamic detection of window starting positions and its implementation within an activity recognition framework

Qin Ni^{a,*}, Timothy Patterson^b, Ian Cleland^b, Chris Nugent^b

^a*Department of Telematic and Electronic Engineering, Universidad Politécnica de Madrid, Spain*

^b*School of Computing and Mathematics, University of Ulster, North Ireland, UK*

Abstract

Activity recognition is an intrinsic component of many pervasive computing and ambient intelligent solutions. This has been facilitated by an explosion of technological developments in the area of wireless sensor network, wearable and mobile computing. Yet, delivering robust activity recognition, which could be deployed at scale in a real world environment, still remains an active research challenge. Much of the existing literature to date has focused on applying machine learning techniques to pre-segmented data collected in controlled laboratory environments. Whilst this approach can provide valuable ground truth information from which to build recognition models, these techniques often do not function well when implemented in near real time applications. This paper presents the application of a multivariate online change detection algorithm to dynamically detect the starting position of windows for the purposes of activity recognition.

Highlights

- Propose a novel data segmentation method based on multivariate online change detection algorithm to dynamic identify a window's starting position.
- Collect accelerometer dataset from a group of participants in a real home scenario.
- Enhance the activity recognition performance by feature selection method and class distribution rebalance.
- Achieve high classification performance of static activities, dynamic activities and transitional activities.

Keywords: Data segmentation, Activities of Daily Living (ADLs), Change detection, Feature selection, Activity recognition.

* Corresponding author.

E-mail address: qin.ni@alumnos.upm.es (Q. Ni), t.patterson@ulster.ac.uk (T. Patterson), i.cleland@ulster.ac.uk (I. Cleland) cd.nugent@ulster.ac.uk (C. Nugent).

1. Introduction

1.1. Background

In the past few decades, there has been a rapid rise of interest in the area of evaluating the routine of elderly members of society with a view to promoting independent living. Activity recognition, particularly the recognition of Activities of Daily Living (ADLs) and ambulatory activities, is an important component in smart home applications for elderly healthcare. One goal of activity recognition is to offer context awareness that allows smart home applications to better understand the user's demands and adapt to the circumstances of the user. In [1] Lockhart *et. al* present a review of application areas for mobile activity recognition (AR). One identified area is fitness tracking where activity recognition algorithms provide opportunity to continuously measure the start/end times and duration of a wide-range of activities. For example, the ActiTracker smartphone based activity system for improving health and well-being [2] identifies activities from the set {'Standing', 'Traversing Stairs', 'Jogging', 'Walking', 'Sitting', 'Lying Down'}. This potential for a high-level of granularity empowers a user to track their day-to-day routine and facilitates algorithms that prompt the user if prolonged sessions of sedentary behaviour are detected. Activity recognition has also been utilised for health monitoring applications such as Parkinson's disease [3] whereby a patient's activity can be tracked longitudinally and examined by a health care professional. Similarly, longitudinal tracking of fine-grained activity facilitates automated actigraphs that can be displayed and interpreted by the end user. For example, in [4] Tryon *et. al* found that children with Attention Deficit Hyperactivity Disorder (ADHD) reduced their activity levels when using positive reinforcement from actigraphs displayed on a wrist worn device. A further use case for activity recognition is within applications that solicit engagement from an end user when a certain condition is satisfied, for example reminders [5] or momentary assessment and experience sampling applications [6].

In terms of diverse technologies for activity detection, accelerometers have received the high attention due to their superiority of long term and accurate activity monitoring. Due to their small size and relatively low cost, accelerometers can be embedded into wrist bands, watches, bracelets and belts to monitor the user's activities and wirelessly send data to mobile computing devices. Most of the research have investigated the use of either a single accelerometer placed on a specific part of body or multiple accelerometers placed on different parts of body [7] [8]. They can measure the value of acceleration along a sensitive axis and are particularly effective in monitoring activities related to body motion. Data from accelerometers usually comes as a continuous data flow of raw data. This is because accelerometers provide instant measurements of the monitored phenomenon, either

when requested or at periodic intervals. To facilitate activity recognition tasks, the continuous data flow needs to be divided into smaller segments for further processing.

One of the main challenges related to accelerometer data segmentation is to achieve a proper division of this raw and continuous data flow into a set of individual segments most suitable for the activity recognition [9] [10]. Each segment is treated as an instance of a specific activity. Segmenting a continuous data flow is a difficult task. On one hand, the user performs consecutive activities probably interleaved with each other rather than separated clearly by pauses. On the other hand, the exact boundaries of an activity are difficult to define. Existing literature has highlighted the sliding-window based segmentation approach for dividing the accelerometer data flow. However, one challenge with this approach is identifying the optimal window size. Since the different types of activities have diverse duration, additionally, transitional activities are usually discarded in most works due to their generally low incidence and very short durations [11].

The overall objective of change detection is to identify locations within an input data stream that exhibit sudden changes in metrics such as mean or variance thus representing a change point in time series data [12]. Depending on the intrinsic characteristics, change detection algorithms can be either classified as online or offline. The aim of online change detection algorithms is to complete processing a set of data points before the next set arrives, with the number of new data points in successive sets influenced by the target application and available computational resources. In the offline case, data is firstly collected and then analysed to detect change points with less emphasis on computational resource requirements. Online change detection algorithms are sequential, fast and minimise false alarms; on the other hand, offline change detection algorithms seek to identify all possible change points in order to attain higher levels of sensitivity (true positives) and specificity (true negatives) [13].

1.2. Related work

The proper selection and parameterization of segmentation techniques has great potential impact on the success of feature extraction and inference algorithms, directly resulting in the accuracy of activity monitoring and recognition. The sliding window segmentation approach is one of the most widely used segmentation techniques for activity monitoring [9]. By using this approach, the continuous sensor data flow obtained by accelerometers are divided into windows with either static or dynamic sizes. In the former case, the fixed window size can be based on equal time intervals. Two segmentation algorithms are of particular interest in current literature: Fixed-size Non-overlapping Sliding Window (FNSW) and Fixed-size Overlapping Sliding

Window (FOSW) [14] [15]. FNSW is a simple segmentation approach without any data overlap, thus the amount of windows can be exactly calculated. However since this algorithm works with a fixed window size, the data associated with a particular event such as a fall could be split in different windows, resulting in important loss of information. FOSW includes data overlap between adjacent windows, and the different overlapping percentages can be referred to as window shifts. [16] set up an experiment to compare the classification accuracy of FOSW with different window overlap values (0%, 25%, 50%, 75% and 90%) applied to acceleration data for recognizing ambulatory activities. Their work determined that the optimal segmentation approach was FOSW with 90% of window overlap and 12 s of window size, reaching a classification accuracy of 98.38%. Dynamic sliding window approach enables varying window sizes based on different sensor features (such as sensor state change or location change of consecutive sensor data) and/or on typical activity duration. Noor et al. [9] proposed a dynamic sliding window method to segment the data obtained by a single accelerometer for monitoring static, dynamic and transitional activities. The window size could be dynamically adjusted by using signal information to determine the most effective segmentation. Krishnan et al. [10] proposed and evaluated a combination of static and dynamic sliding window sizes. They improved the sliding window approach with three modifications, which oriented to calculate the size of the window to be used at each moment depending on the sensor data, the environment and the monitored activity, to capture the relationship between sensor events within a window and between multiple windows.

One challenge in sliding window segmentation approach is to optimize the window size by considering both short and long duration activities. In current literature, a wide range of window sizes are adopted depending on the different monitored activities (shown as Table 1). Fida et al. [10] presented the impact of window size on the recognition of both short duration, such as sitting, standing and transitions between activities, and long duration activities, such as walking. They found that 1.5 s window size represented the best trade-off in recognition among activities. Banos et al. [17] evaluated the effect of different window sizes on 33 different fitness exercises recognition with a non-overlapping sliding windowing approach. These window sizes ranged from 0.25 s to 7 s in steps of 0.25 s. As a result they proved that short windows normally resulted in better recognition performance, and the interval 1 s to 2 s exhibits the best trade-off between recognition speed and accuracy from a global perspective. In general the sliding window-based approach is appropriate for data streams that are continuously obtained by sensors over a relatively long period of time. Furthermore, it has the advantages in terms of less computation complexity in comparison to other methods. Nevertheless, the selection of the optimal value for the time interval is a critical issue. Too small an interval may split one activity into two

adjacent windows, especially for a long duration activity such as cooking, resulting in a lack of sufficient information to arrive to an accurate classification. On the other hand, a too wide interval may gather two or more activities into the same segment and add extra noise for further data processing. Thus, the novelty of this paper lies in dynamically detecting *changes* in a user's activity and subsequently utilizing these transitions in activity as window starting positions. For activities with durations greater than the window length this approach helps to ensure that each window only contains sensor values for a single activity.

Table 1. A range of different window sizes used in current literature.

Reference	Sensors	Monitored activities	Window size	Classification accuracy
Kwapisz et al. [18]	1 accelerometer	Walking, jogging, ascending stairs, descending stairs, sitting and standing	10s	91.7%
Siirtola et al. [19]	1 accelerometer	walking, running, cycling, driving a car, sitting and standing	7s	95.8%
Jiang et al. [20]	4 accelerometers	Standing, sitting, lying, walking, jogging, cycling, walking, running, rowing, weight lifting	6s	95.1%
Chernbumroong et al. [21]	1 accelerometer, temperature sensor, altimeter	Feeding, brushing teeth, dressing, walking, walking upstairs, walking downstairs, sleeping, washing dishes, ironing, sweeping, watching TV	3.88s	90.2%
Ortiz et al. [11]	1 accelerometer	Standing, sitting, lying, walking, walking downstairs and walking upstairs, stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand	2.56s	92.2%
Gao et al. [22]	1 accelerometer	Lying, sitting, standing, walking, lying-standing, standing-lying, sitting-standing, and standing-sitting.	1s	92.8%

Given our target application of activity recognition on a wearable device we focus on online change detection algorithms throughout the remainder of this paper. In the literature there are a number of algorithms for online change detection in sensor data. One such example is the Cumulative Sum Control Chart (CUSUM) which utilises the mean of a process for detecting small shifts. In [23] Zhang *et al.* presented a framework for detecting changes in cardiovascular events using CUSUM. The core methods used in the framework were an online AR method, a biometric extraction method and a process control method used by the physiological monitoring module. A criticism of CUSUM was, however, that it does not robustly detect sudden shifts in data that are not from the same distribution [24] and therefore may be unsuitable for change detection based on accelerometer data. The univariate change detection algorithm proposed by Jain and Wang (2014) [25] was used to detect changes in independent random sequences. The algorithm consisted of two stages: in the first phase the most likely change point within a processing window was identified. In the second stage the hypothesis that the most

likely change point was significant was proved or disproved. The main advantage of the algorithm is that it does not require knowledge of the underlying distribution, has a small memory footprint and a relatively low computational cost. In [26] and [27] we presented the Multivariate Online Change detection Algorithm (MOCA) that has been developed to detect transitions in a multivariate input data stream. MOCA does not require knowledge of the underlying component distributions, can operate on multivariate data streams and can execute at approximately 28Hz on a smartphone.

2. Data Collection

Three categories of activities are taken into consideration in this study: static activities, such as standing and sleeping, dynamic activities, such as walking and running, and transitional activities, such as stand-to-sit and sit-to-stand. Transitional activities commonly occur within a limited duration, that are characterized by start and end times with the duration varying between individuals [28]. Subsequently, static activities and dynamic activities can be executed continuously, whilst transitional activities can be executed repeatedly to obtain sufficient samples.

Ten healthy adults (3 females and 7 males) were recruited to participate in this study from the Ulster University. The participants were asked to perform the 12 activities according to the experimental protocol in a controlled laboratory environment. A tri-axial accelerometer was attached to each participant's left wrist by using a custom wrist-belt. Previous research has shown sensors on the wrist provided reasonably good rates of precision and recall for some fine grain activities [29]. Furthermore, the wrist provides a comfortable location at which to wear the device and is a common location for smartwatches and activity monitors. The Shimmer wireless sensor platform (Shimmer 3, Realtime Techniques, Dublin, Ireland) was used to record the calibrated accelerometer data due to its sufficiently small and lightweight form factor (53mm*32mm*25mm in size and 22g in weight). This ensures that it can be comfortably worn by the participant and is well suited for long-term wearable activity monitoring applications. Prior to use, the device was calibrated according to the manufacturer guidelines.

Within this work we considered 12 common activities: standing, sleeping, watching TV, walking, running, sweeping, stand-to-sit, sit-to-stand, stand-to-walk, walk-to-stand, lie-to-sit and sit-to-lie. Table 2 shows the classification and description of the twelve activities into three types. The experimental protocol consisted of asking participants to perform a guided sequence of these activities by using voice instruction. Participants were told the activity to perform but not specifically how to perform it, thus this experimental protocol allowed for

natural participant variability in how activities are performed. For stationary activities, each participant was asked to perform standing still, sleeping on the sofa and watching TV. Participants could choose a comfortable posture with small movements such as changing lying posture during sleeping and changing sitting posture during watching TV allowed. For dynamic activities, each participant was asked to perform walking on treadmill with preferred speed, running on treadmill with preferred speed and sweeping with the vacuum cleaner in the home area. All activities were performed for 5 minutes each. For transitional activities, each participant was asked to perform the three pairs of activities (stand-to-walk-to-stand, stand-to-sit-to-stand and sit-to-lie-to-sit), each pair is performed for 45s, then repeated 15 times. The protocol took approximately 1 hour for each participant to complete.

Table 2. The taxonomy and description of monitored activities.

Type	Activities	Description
Stationary Activities	Standing	Standing still for 5 minutes
	Sleeping	Sleeping on the sofa for 5 minutes, small movements such as changing the lying posture , are allowed
	Watching TV	Watching TV while sitting on the sofa in whatever posture the participant feels comfortable for 5 minutes, changing sitting posture is allowed
Dynamic Activities	Walking	Walking on treadmill with the set speed for 5 minutes
	Running	Running on the treadmill for 5 minutes
	Sweeping	Sweeping with the vacuum cleaner in the home area for 5 minutes
Transitional Activities	Stand-to-sit	Standing still for 15s and then sitting on the sofa, repeat for 15 times
	Sit-to-stand	Sitting on the sofa for 10s and then standing up, repeat for 15 times
	Stand-to-walk	Performing the “stand-to-walk-to-stand”, standing still for 15s then start to walk, keep walking for 15s, then standing still for 15s, repeat for 15 times
	Walk-to-stand	
	Lie-to-sit	Lying on the sofa for 15s and then sitting on the sofa, repeat for 15 times
Sit-to-lie	Sitting on the sofa for 15s and then lying down, repeat for 15 times	

Calibrated acceleration data was collected with a sampling rate of 102.4 Hz. Consistent sampling allowed time domain and frequency domain signal analysis. Previous studies have demonstrated that human movements can be captured and modeled by signals below 18 Hz [30]. Thus a sampling rate of 102.4 Hz was considered to be more than sufficient for capturing transitional activities as well as activities of daily living. The sensed data were stored in memory card embedded into the Shimmer wireless sensor platform, the activity data of each participant were then exported in a separate spreadsheet file to a notebook computer, were saved for offline analysis. The collected data were in the format of {timestamp, acceleration-X, acceleration-Y, acceleration-Z}. As a result, over ten hours of raw dataset, which contained 4,020,288 instances, were acquired.

The collected data was annotated during the activities execution by using the voice instruction and labelling the start and end time of an observed activity manually. Given the short duration of the transitional activities and the potential for a delay in the participant performing the activity following the verbal command, the following labeling procedure was undertaken for transitional activities. The duration of transitional activities has been proven in the time interval from 1.45 seconds to 3.49 seconds [31]. Taking the different reaction time of participants into consideration, 4s was used to label the transitional activities in order to cover the whole transitional phase. Consequently, 2160 seconds of stable state data was labeled with twelve activities in the raw data. These labeled data can be used to calculate training or testing feature vectors later.

3. Change Point Detection

In the context of this work the aim of change detection is to identify transitions in a user's activity thus determining the starting position of windows to be utilized in window-based activity classification. In this Section we provide an overview of MOCA and how the detected changes are used within the overall activity recognition framework. MOCA autonomously detects changes in an input data stream such as accelerometry data as follows: consider a data stream of length q consisting of data points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$. Each data point \mathbf{x} is a b element vector where b is the number of sensor observations for each variable. The data stream may contain points from multiple distributions, for example $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{k-1}$ may have distribution D_1 whilst $\mathbf{x}_k, \mathbf{x}_{k+1}, \dots, \mathbf{x}_q$ may have distribution D_2 . It is therefore the overall aim of the algorithm to identify the position in the data stream of change points k ($1 \leq k \leq q$)

MOCA follows an hypothesis-and-verification principle: in the hypothesis step a point is detected within the window under consideration which maximises the test statistic. In the second stage the hypothesis that a detected change point is significant is verified.

3.1 Hypothesis Generation

In the hypothesis generation stage we pass an analysis window of length n over the data stream assuming that there is a maximum of one change point per window. The movement of the window over the data stream may be either distinct in which case the start of a new window (other than the first) is at position $m + cn + 1$ where m is the padding size and c is the number of previous windows. Alternatively, a sliding window version of the algorithm may be executed with the start position incremented by a predetermined number of data points. For ease of notation we denote the data points within a window as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ regardless of their actual

position within the data stream. Following Jain and Wang [25] we pad either side of the window with m data points such that the analysis window contains data points $\mathbf{x}_{1-m}, \dots, \mathbf{x}_{n+m}$ therefore containing a total of $n + 2m$ data points. This padding ensures sufficient data to compute summary statistics at the window extremities and is particularly crucial when executing a distinct window version of the algorithm; however, results in a minimum latency of m data points.

Within each window we slide an index variable, $l, 1 < l \leq n$ subsequently computing summary statistics of the component distributions separated at l . Specifically, we compute the means, $\bar{\mathbf{f}}_1(l)$ and $\bar{\mathbf{f}}_2(l)$, which contain the mean of observations, in addition to variance-covariance matrices, $\mathbf{S}_1(l)$ and $\mathbf{S}_2(l)$, which contain the variance of observations in the diagonals and their covariance in the off-diagonals. To ensure that the change detection algorithm can operate in online scenarios we compute $\bar{\mathbf{f}}_1(l)$, $\bar{\mathbf{f}}_2(l)$ and $\mathbf{S}_1(l)$, $\mathbf{S}_2(l)$ recursively. Thus as index l increments to position $l + 1$ the summary statistics are calculated as follows:

$$\bar{\mathbf{f}}_1(l + 1) = \frac{m+l-1}{m+l} \bar{\mathbf{f}}_1(l) + \frac{f(\mathbf{x}_{l+1})}{m+l}, \quad (1)$$

$$\bar{\mathbf{f}}_2(l + 1) = \frac{n+m-l+1}{n+m-l} \bar{\mathbf{f}}_2(l) - \frac{f(\mathbf{x}_{l+1})}{n+m-l}, \quad (2)$$

$$\mathbf{S}_1(l + 1) = \frac{m+l-1}{m+l} \mathbf{S}_1(l) + \frac{1}{m+l-1} \mathbf{X} [\mathbf{x}_{l+1} - \bar{\mathbf{f}}_1(l + 1)]' [\mathbf{x}_{l+1} - \bar{\mathbf{f}}_1(l + 1)], \quad (3)$$

$$\mathbf{S}_2(l + 1) = \frac{n+m-l+1}{n+m-l} \mathbf{S}_2(l) - \frac{1}{n+m-l} \mathbf{X} [\mathbf{x}_{l+1} - \bar{\mathbf{f}}_2(l + 1)]' [\mathbf{x}_{l+1} - \bar{\mathbf{f}}_2(l + 1)]. \quad (4)$$

where $f(\mathbf{x}_{l+1})$ is the value of the datastream at position $l + 1$. Having calculated summary statistics before and after l we proceed to compute the F statistic at position l , F_l as follows [32]:

$$F_l = \frac{n_1 + n_2 - b - 1}{b(n_1 + n_2 - 2)} T^2, \quad (5)$$

where $n_1 = m + l - 1$, $n_2 = n + m - l + 1$, b is the number of variables and T^2 is the Hotelling T-squared statistic calculated as [32],

$$T^2 = (\bar{\mathbf{f}}_1 - \bar{\mathbf{f}}_2)' \left\{ \mathbf{S}_p \left(\frac{1}{n_1} + \frac{1}{n_2} \right) \right\}^{-1} (\bar{\mathbf{f}}_1 - \bar{\mathbf{f}}_2), \quad (6)$$

where \mathbf{S}_p is the pooled variance-covariance matrix,

$$S_p = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1 + n_2 - 2}. \quad (7)$$

Under the null hypothesis (i.e. equal distributions) and assuming Gaussian distributions this has a F distribution [32]. We choose the point l which maximises F_l as the most likely change point within a window and proceed to the hypothesis verification phase.

3.2 Hypothesis Verification

A hypothesis verification stage is executed to prove or disprove the null hypothesis that a significant change did not occur at point l . Firstly, we compute the probability of finding a F value lower than that calculated in Equation 5 resulting in d . The F Cumulative Distribution Function is utilized for this phase with b and $n_1 + n_2 - b$ degrees of freedom. The test's p -value is then computed as,

$$p = 1 - d. \quad (8)$$

As multiple statistical tests are being simultaneously performed within the window it is necessary to adjust our confidence value, α to reflect the confidence for the entire window and not a single, isolated value. We therefore use a Bonferroni correction [33] to compute an adjusted threshold t as:

$$t = \frac{\alpha}{n}, \quad (9)$$

with the null hypothesis rejected (i.e. a significant change did occur at l) if $p < t$.

4. Activity recognition methods

As shown in Figure 1, the activity recognition framework is a sequence of data processing and machine learning techniques. Firstly, the collected raw accelerometer data are preprocessed to filter out the unwanted noise and separate the body acceleration from gravity acceleration. The processed data are then segmented into windows by following a two-step data segmentation approach, each window in the dataset is extracted as an input of 77 features by the feature extraction process. Based on this, the feature selection method is used to select the optimal feature subset by choosing relevant features and removing redundant features. The resampling

technique is applied to rebalance the class distribution in the dataset. Afterward, the extracted features with the corresponding ground truth activity labels are used as input to train and test the classifier model.

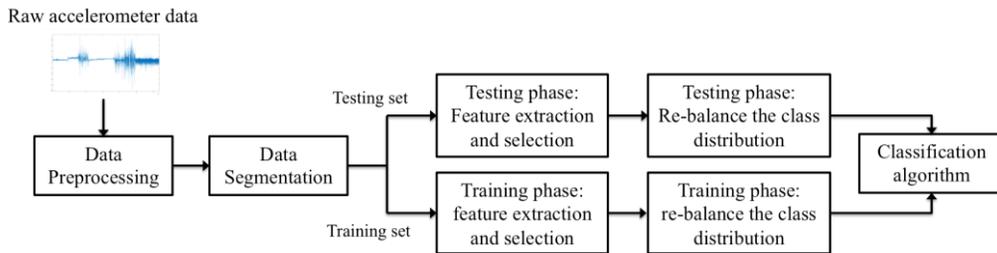


Fig. 1. The conceptual scheme of the activity recognition framework.

4.1. Data preprocessing

Generally, the acceleration data collected in real-world scenarios contain noise. This noise may be introduced by either external vibration not produced by the body itself, or by movement of the sensor against the body due to loose coupling. Since essentially all measured body motions are contained within frequency components below 20 Hz [34], a third-order low-pass Butterworth filter with cutoff frequency at 20 Hz was used to remove the high-frequency noise and smooth the graphs. After that, the acceleration data were filtered by a third-order median filter to remove abnormal noise spikes and solve signal delayed problem caused by the low-pass Butterworth filter [11].

Additionally, the acceleration data collected from the accelerometer were decomposed into body acceleration (BA) component, which caused by the body movement that can be used to distinguish motion activities from the static activities, and gravity acceleration (GA) component, which caused by the gravity that can be used to estimate the posture orientation of a participant when performing the activities. In order to separate the BA component and GA component from the filtered signal, a third-order high-pass Butterworth filter with cutoff frequency at 0.3 Hz was applied to extract the BA component from the filtered signal. Afterward, the GA component was obtained as the difference between the filtered signal and the BA component.

4.2. Data Segmentation

As shown in Fig. 2, we adopted a two-step approach for accelerometer data segmentation. The objective of the proposed approach is to map a set of accelerometer data samples with a corresponding activity label to the most accurate extent. In step 1, as specified in Section 3, let $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$ represented the data samples in the accelerometer dataset collected from smart home. MOCA identified significant changes in the input data stream. The results of MOCA were a set of change point indices, C_1, C_2, \dots, C_n $n \leq q$. Formally, the data samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$ were divided into segments of S_1, S_2, \dots, S_n , $n < a$, where according to the indexes of change points, and the S_i segment contained the data samples $[\mathbf{x}_{c_i}, \dots, \mathbf{x}_{c_{i+1}-1}]$.

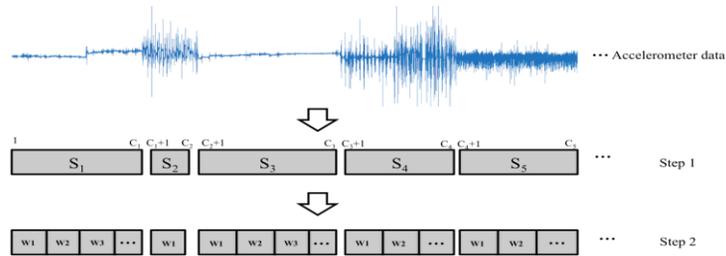


Fig. 2. Illustration of the proposed segmentation method based on change point detection.

In order to reduce the information loss, in step 2, the S_i segment was divided into windows of 256 samples with a 50% overlap between two consecutive windows, which represented by w_1, w_2, \dots, w_N . Therefore, each classification task made about the activity was performed for the duration of 2.5 seconds windows. Once the window w_i in the segment S_i was defined, the next step was to compute and transform this window into a feature vector.

4.3. Feature extraction and selection

A feature vector consists of both time-domain features as well as frequency-domain features was calculated on each windowed data. The extracted features were obtained by processing the acceleration values of X-, Y-, and Z-axis of both BA component and GA component. The signal magnitude vector (SMV), which defined as $\sqrt{x^2 + y^2 + z^2}$ was also used to extract features. Since the SMV provides a measure of the degree of body movement intensity, which can be calculated by using the tri-axial acceleration values [35].

Commonly used time-domain statistical features, such as mean, root mean squared, standard deviation and correlation coefficient were also used in this work [36]. Additionally, the signal magnitude area (SMA) and tilt angle were included in our study [37]. The SMA computes the energy expenditure in activities by using the sum of area encompassed by the magnitude of three-axis acceleration, therefore it can be used to distinguish a rest state and dynamic activity. The tilt angle provides the information on the determination of the participant orientation, which is useful for transitional activities classification. The extracted frequency-domain features included spectral energy, spectral entropy, skewness, kurtosis, largest frequency component and signal weighted average. The list of 77 features used in our study was summarized in Table 3.

Table 3. Features extracted from time and frequency domain.

Features domain	Features	Data values	Total number of features
Time-domain features	Mean		
	Root mean squared (RMS)	BA-X, BA-Y, BA-Z, GA-X, GA-Y, GA-Z, BASMV, GASMV	40
	Standard deviation (STD)		
	Median absolute deviation (MAD)		
	Range		
	Signal magnitude area (SMA)	BA, GA, BASMV, GASMV	4
	Correlation coefficient	BA-XY, BA-YZ, BA-XZ, GA-XY, GA-YZ, GA-XZ,	6
	Tilt angle (TA)	BA-X, BA-Y, BA-Z	3
Frequency-domain features	Spectral energy		
	Spectral entropy	BA-X, BA-Y, BA-Z, BASMV	24
	Skewness		
	Kurtosis		
	Largest frequency component		
	Frequency signal weighted average		

Within the data, there existed situations when more than one activity label fell into the same segmented window. When dealing with multiple activities or less than 256 samples within a window, this window was discarded. As a result, a feature vector x_i was tagged with the 77 features and the corresponding activity label y_i . Then a set of feature vectors x_i and the corresponding activity labels y_i were fed into a classifier as the training data and testing data.

Features extracted from calibrated sensor data may contain redundant and irrelevant information, which can negatively affect classification performance. In order to identify a subset of the most discriminative features that can increase the classification performance, as well as remove the redundant features that contribute no

additional information to the classifier, it is essential to apply an effective feature selection method. There are two main categories of feature selection methods: filter methods and wrapper methods [38]. In this study, three filter feature selection methods were considered: Information Gain (InfoGain), Correlation-based Feature selection (CFS) with Best First search (BFS) algorithm and ReliefF. We chose these three statistical filter methods because the filter methods for feature selection rely on general data characteristics rather than the learning algorithms, while the wrapper methods require predetermined classifiers and their performance quite depend on the classifiers [39].

4.4. Class distribution re-balance

Due to the relatively short durations of transitional activities compared to static activities and dynamic activities, there are much less instances of transitional activities than of the other classes. Due to the imbalance, there is the potential of having a bias towards the majority class. Leading to higher misclassification rate for the minority class instances. This impacts upon the overall accuracy of the classification. In order to deal with the problem of class distribution imbalance in the dataset, the resampling technique was integrated in this study to rebalance the class distribution. Previous works have reported that, randomly over-sampling methods would result in over-fitting problem and under-sampling methods would lead to the loss of useful information [40]. Inspired by the work demonstrated in [41] [42], which suggested the SMOTE technique that generates synthetic instances along the windows adding any of the k minority class to their nearest neighbors, and combination of over-sampling technique on the minority class and under-sampling technique on the majority class, we utilised three different resampling techniques on the dataset: SMOTE boosting method, which the percentage of data to create for six transitional activities classes with less number of instances was set to 200%, and the combination of over-sampling and under-sampling techniques with a 2:1 ratio and 3:1 ratio of the majority class samples to minority class samples. These parameters have shown as optimum within previous works for rebalancing datasets [41] [42]. The performance of each activity class obtained from three techniques were then compared to determine the optimal balanced class distribution.

5. Results and Discussion

In this section, we present and discuss the results obtained from segmentation approaches, feature selection methods, resampling techniques and learning algorithms applied on the dataset. To precisely assess the

performance evaluation and model comparison, several performance measurements, such as accuracy, true positive rate, false positive rate and F-measure, are used.

5.1. Performance improvement using proposed segmentation method

In order to indicate the classification performance improvement of the proposed segmentation method, the experiment was designed in two phases. In the first phase, the preprocessed dataset was segmented by using three different segmentation methods: the common used non-overlapping segmentation method and 50% overlapping segmentation method as well as our proposed segmentation method based on change point detection. Both non-overlapping segmentation method and 50% overlapping segmentation method adopted 2.56 s as the window size. Then 77 features mentioned before were extracted from each window. As a result, three different inputs, which consisted of 77 features and activity labels were obtained. In the second phase, in order to choose the most efficient classifier for the classification, we investigated and compared various classification algorithms: random forest (RF), k-nearest neighbor (KNN), decision tree (J48), artificial neural network (MLP), naïve bayes (NB) and support vector machine (SVM). For each input, 10 runs of 10-fold cross validation were executed. For each iteration of cross validation, nine folds served as training datasets and the remaining fold served as the test dataset.

Table 4. Performance evaluation for the proposed segmentation approach based on change point detection.

	J48	RF	KNN	MLP	NB	SVM
Non-overlapping segmentation	90.05	93.56 •	93.40 •	92.97 •	69.38 ◦	19.55 ◦
50% overlapping segmentation	92.40	94.71 •	93.73 •	91.31	69.81 ◦	32.72 ◦
Change detection-based segmentation	93.11	95.53 •	94.12	93.74	79.15 ◦	60.94 ◦

• statistically significantly better, ◦ statistically significantly worse, level of significance 0.05

The average accuracy of six different algorithms on three inputs from the experimentation were tabulated in Table 4. Here the performance of J48 was used as the baseline for comparison. The table showed how often each algorithm (column) performed statistically significantly better (denoted by a •) or worse (denoted by a ◦) than the baseline for the respective input (row) according to a paired two-sided t-test. From Table 4, it can be seen that the classification results of NB and SVM were not good, even statistically worse than the baseline. RF showed statistically significantly better than the baseline, contributing the highest accuracy of 95.53% on the input obtained by using change detection-based segmentation, which was even higher than the highest accuracy obtained from [10]. This probably because RF combined multiple decision tree with various discriminative

rules, which can handle the multi-class data characteristics. Compared to RF, KNN, MLP and J48 contributed slightly lower classification accuracy of 94.12%, 93.74% and 93.11%, respectively. Therefore, it was decided to use only RF in this study.

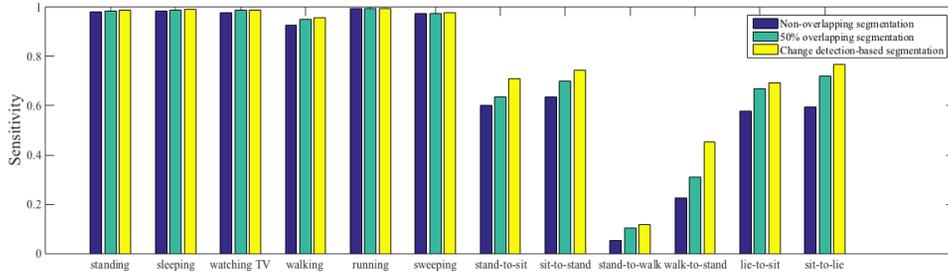


Fig. 3. The sensitivity of each activity class of three segmentation methods.

In terms of each input, the average accuracy of non-overlapping segmentation method, 50% overlapping segmentation method and change detection-based segmentation method by applying RF algorithm were 93.56%, 94.71% and 95.53%, respectively. Figure 3 showed the sensitivity of each activity class obtained by applying RF algorithm on three inputs. Both non-overlapping segmentation method and 50% overlapping segmentation method generally performed well in classifying static activities and dynamic activities but performed less well in classifying transitional activities. This may be due to the duration of transitional activities being typically short, each window obtained by using non-overlapping segmentation method and 50% overlapping segmentation method may contain more than one activities rather than a single transitional activity, which resulted in misclassification of the transitional activities.

The results indicated that change detection-based segmentation method contributed better performance in terms of either overall classification accuracy or the specific classification accuracy of transitional activities. This is because change detection-based segmentation method reorganized windows better and more effectively. However, compared to static activities and dynamic activities, change detection-based segmentation method still performed relatively bad in classifying transitional activities. Thus, in the next steps, we adopted the feature selection methods and resampling techniques to enhance the performance on transitional activities classification.

5.2. Feature Selection enhancing the performance

The optimal feature subset selected by the feature selection method should contain most relevant and discriminative features, and the number of features should be as small as possible to reduce the complexity of learning process. In order to select the most appropriate feature selection method, the effectiveness of the 10-fold cross validation performance by running 10 iterations of the three filter methods was compared in two groups. The performance of using all features was served as the baseline for comparison. Five feature subsets were used: FS1 using InfoGain with 60 selected features, FS2 using ReliefF with 70 selected features, FS3 using CFS and BFS algorithm with 21 selected features, FS4 using InfoGain with 21 selected features, FS5 using ReliefF with 21 selected features. Each group consisted of three feature subsets: FS1, FS2 and FS3 belonged to group 1. FS3, FS4 and FS4 belonged to group 2. It should be noted that all features are ranked by InfoGain method and ReliefF method based on different criteria, while the feature subset was picked directly by CFS and BFS algorithm. The average 10-fold cross validation performance of five feature subsets was shown in Table 5. The RF was used here as machine learning classifier for classification.

Table 5. Comparison of the performance obtained by three feature selection methods.

	No. of features	Accuracy	F_measure	ROC area
All features	77	95.53	0.98	1.00
FS1: InfoGain	60	95.37	0.98	1.00
FS2: ReliefF	70	95.57	0.98	1.00
FS3: CFS+BFS	21	95.62 •	0.98	1.00
FS4: InfoGain	21	93.22 ◦	0.96 ◦	0.99 ◦
FS5: ReliefF	21	95.24	0.98	1.00

- statistically significantly better, ◦ statistically significantly worse, level of significance 0.05

In group 1, The CFS and BFS algorithm directly selected a feature subset (FS3) that contained 21 features and provided the highest accuracy of 95.62% among three feature subsets. Compared to the performance of using all features, the classifier produced higher accuracy and showed statistically significantly better with only 21 features. This is due to the fact that some features are irrelevant and redundant, a high correlation with other features would affect the performance. The feature selection methods selected only informative and relevant features that contributed to improve the classification performance.

To determine the optimal feature subset, the performance of three feature subsets with the same number of features by three feature selection methods were compared (group 2). With the number of used features decreased, the top 21 ranked features by InfoGain only produced an accuracy of 93.22% and statistically significantly worse than the baseline. Similarly, the accuracy with top 21 ranked features by ReliefF decreased.

Among the five different feature subsets, the CFS and BFS algorithm yielded the highest accuracy and best performance. Therefore, the CFS and BFS algorithm was used as the feature selection method to reduce the feature vector dimensions as well as improve the classifier performance. The 21 picked features by the CFS and BFS algorithm were given in Table 6.

Table 6. The 21 features selected by CFS and BFS algorithm.

Feature Domain	Feature Number	Selected Feature name (Axis applied to)
Time Domain	1	STD- (BA-X)
	2	MAD (BA-X)
	3	Correlation (BA-XY)
	4	Tilt (BA-X)
	5	Tilt (BA-Y)
	6	Tilt (BA-Z)
	7	Mean (GA-Y)
	8	Mean (GA-Z)
	9	RMS (GA-X)
	10	RMS (GA-Y)
	11	Correlation (GA-XZ)
	12	SMA (GA)
	13	Mean (BA-SMV)
	14	MAD (BA-SMV)
	15	STD (GA-SMV)
Frequency Domain	16	Kurtosis -BA-X
	17	Kurtosis -BA-Y
	18	Kurtosis -BA-Z
	19	Largest frequency component-BA-X
	20	Largest frequency component-BA-Y
	21	Frequency signal weighted average-BASMV

5.3. Class distribution re-balance enhancing the performance

We evaluated the classification performance of RF classifier on the selected features by using three different resampling techniques: SMOTE, combination with 2:1 ratio and combination with 3:1 ratio. The classification performance of imbalanced dataset was used as the baseline. Besides the overall classification accuracy, we additionally used true positive rate (TPR), false positive rate (FPR), F-measure and ROC area as the performance measurements to capture classification performance for both majority and minority classes. As shown in Table 7, among the performance obtained from three resampling techniques, SMOTE yielded the highest accuracy of 95.76% and standard deviation of 0.53, slightly higher than the baseline. The performance of the combination with 2:1 ratio and the combination with 3:1 ratio both showed the statistically significantly

worse than the baseline in terms of accuracy, TPR and F-measure. Since the SMOTE technique did not show the statistically significantly better than the baseline in terms of overall evaluation performance, the sensitivity of each activity class was analyzed later to determine if the SMOTE technique can enhance the performance.

Table 7. The classification performance by using three resampling techniques.

	Imbalanced dataset	SMOTE	Combination with 2:1 ratio	Combination with 3:1 ratio
Accuracy	95.62	95.76	90.84 ◦	90.92 ◦
TPR	0.98	0.98	0.94 ◦	0.94 ◦
FPR	0.00	0.00	0.00	0.00
F-measure	0.98	0.98	0.95 ◦	0.95 ◦
ROC area	1.00	1.00	1.00	1.00

- statistically significantly better, ◦ statistically significantly worse, level of significance 0.05

Figure 4 depicted the sensitivity of three resampling techniques on each activity class. Compared to performance without using the resampling technique, all three resampling techniques increased the sensitivity of transitional activity classes that were balanced with the static activity classes and dynamic activity classes. Among them, the sensitivity of static activities and dynamic activities decreased by using combination with 2:1 ratio and combination with 3:1 ratio. However, SOMTE contributed more or less the same sensitivity of static activities and dynamic activities and on the other hand, increased the sensitivity of six transitional activities. Particularly the sensitivity of sit-to-stand and stand-to-sit activities were increased significantly by using SMOTE.

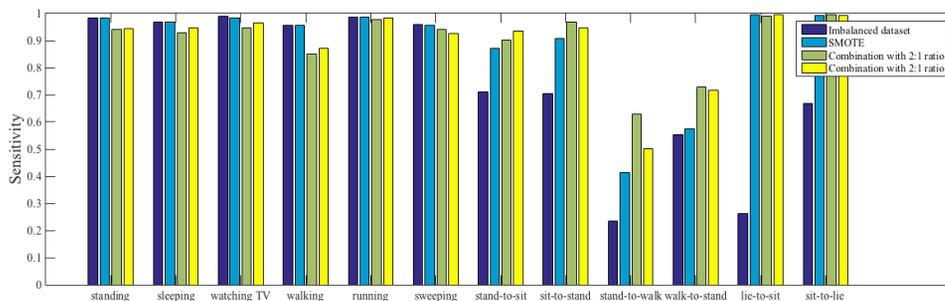


Fig. 4. Comparison of sensitivity of twelve activities by applying three resampling techniques.

5.4. Discussion on the proposed model

In order to indicate the improvement of our proposed model, we compare the performance of three models (M1, M2 and M3) by considering the sensitivity of each activity class. M1 was obtained by applying non-overlapping segmentation method on the collected raw dataset with all features, M2 was obtained by applying

50% overlapping segmentation method on the collected raw dataset with all features and M3 was obtained by applying the proposed change detection-based segmentation method on the balanced dataset with selected features. The classification performance of these three models were given in Table 8.

Table 8. Comparison of sensitivity of each activity class obtained from three models.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Average
M1	0.983	0.985	0.978	0.926	0.995	0.975	0.600	0.637	0.054	0.226	0.578	0.596	0.936
M2	0.985	0.990	0.986	0.947	0.994	0.975	0.642	0.689	0.086	0.279	0.654	0.692	0.945
M3	0.982	0.968	0.984	0.957	0.986	0.956	0.871	0.908	0.214	0.488	0.995	0.992	0.957

A1= Standing, A2= Sleeping, A3= Watching TV, A4= Walking, A5= Running, A6= Sweeping, A7= Stand-to-sit, A8= Sit-to-stand, A9= Stand-to-walk, A10= Walk-to-stand, A11=Lie-to-sit, A12= Sit-to-lie.

From Table 8, it is can be seen that M3 achieved the average sensitivity of 0.957, higher than the average sensitivity of M1 (93.6%) and M2 (94.5%). In terms of each activity class, M3 achieved high performance on static activities classification and dynamic activities classification, activities such as standing, watching TV and running also were classified correctly very well. Moreover, compared to M1 and M2, the proposed model also achieved significant improvement on transitional activities classification. It is worth to notice that the proposed model performed extremely well in classifying sit-to-lie and lie-to-sit activities, achieving the sensitivity as high as 0.995 and 0.992, respectively. Even though M3 improved the sensitivity of sit-to-walk and walk-to-stand activities compared to M1 and M2, while considering other transitional activities, the proposed model did not perform very well in sit-to-walk and walk-to-stand activities classification, only achieved the sensitivity of 0.214 and 0.488, respectively.

Table 9. The confusion matrix obtained by using RF on the balanced dataset with selected features.

Classified as	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
A1	1833	0	10	3	0	16	0	0	0	5	0	0
A2	0	1950	34	16	1	11	0	0	0	0	1	2
A3	1	10	2064	8	0	5	3	0	0	1	2	4
A4	1	4	27	2175	0	54	3	5	0	2	1	1
A5	1	7	7	12	2252	6	0	0	0	0	0	0
A6	11	0	5	46	1	1533	0	2	2	2	2	0
A7	2	0	2	0	0	6	81	1	0	1	0	0
A8	0	0	2	1	0	7	0	139	0	2	2	0
A9	4	0	2	88	0	8	0	1	31	10	0	1
A10	25	0	12	25	0	12	1	4	3	81	1	2
A11	0	1	0	0	0	0	0	0	0	0	206	0
A12	0	0	0	0	1	0	0	0	0	0	1	241

A1= Standing, A2= Sleeping, A3= Watching TV, A4= Walking, A5= Running, A6= Sweeping, A7= Stand-to-sit, A8= Sit-to-stand, A9= Stand-to-walk, A10= Walk-to-stand, A11=Lie-to-sit, A12= Sit-to-lie.

Table 9 showed the confusion matrix of each activity class of the proposed model, which obtained by using RF on the balanced dataset with selected features. The confusion matrix illustrated that the stand-to-walk class was mostly often classified as walking class (88 instances), the walk-to-stand class was confused with standing class (25 instances) or walking class (25 instances). This is because these activities shared quite similar movements and the transition duration was quite short. Walking class was also frequently misclassified as sweeping class (54 instances) by the model. The combination of multimodal sensor data from additional sensors such as gyroscope or magnetometer data could be useful in further discriminating between these very similar activities.

6. Conclusion

In this paper we have presented an approach to windowing that utilises an autonomous, multivariate online change detection algorithm to dynamically identify a window's starting position such that it contains homogeneous activity types. The data segmentation based on change detection algorithm is for the purposes of activity recognition from body worn sensors. In addition, with the discriminative features and balanced class distribution, the transitional activities can be classified correctly well, which are often ignored in current research. This is particularly important for the application of activity recognition in real time.

Results are presented within the context of an activity recognition framework that demonstrate statistically significant performance gains when using this approach in comparison to non-overlapping segmentation method and 50% overlapping segmentation method as commonly used in the literature. There will be three main facts of future work: firstly, we will evaluate our approach against alternative activity recognition frameworks and apply this change detection-based segmentation on more available datasets; secondly, we will evaluate the algorithm in a real-time real-world deployment potentially using smartphones. It is envisaged that with further refinement and evaluation the algorithm presented in this paper will prove to be an invaluable component of applications requiring robust, real-time activity recognition. Thirdly, we will consider multimodal sensor streams to better differentiate between similar activities.

References

- [1] Lockhart JW, Pulickal T, Weiss GM. Applications of Mobile Activity Recognition. In: Proceedings of the 14th ACM International Conference on Ubiquitous Computing; 2012 Sep 5-8; New York(USA). ACM Press; 2012. p 1054–8.
- [2] Weiss G, Lockhart J, Pulickal T, McHugh P, Ronan I, Timko J. Actitracker: a smartphone-based activity recognition system for improving health and well-being. ACM SIGKDD Exploration Newsletter 2014.

- [3] Albert MV, Toledo S, Shapiro M, Kording K. Using mobile phones for activity recognition in Parkinson's patients. *Frontiers in Neurology* 2012;158(3).
- [4] Tryon WW, Tryon GS, Kazlauskas T, Gruen W, Swanson JM. Reducing Hyperactivity with a Feedback Actigraph: Initial Findings. *Clinical Child Psychology and Psychiatry* 2006;11(4):607–17.
- [5] Feuz KD, Cook DJ, Rosasco C, Robertson K, Schmitter-Edgecombe M. Automated Detection of Activity Transitions for Prompting. *IEEE Transactions on Human-Machine Systems* 2015;45(5):575–85.
- [6] Cleland I, Han M, Nugent C, Lee H, McClean S, Zhang S, Lee S. Evaluation of prompted annotation of activity data recorded from a smart phone. *Sensors* 2014;14(9):15861–79.
- [7] Mannini A, Intille SS, Rosenberger M, Sabatini AM, Haskell W. Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in Sports and Exercise* 2013; 45(11):2193.
- [8] Cleland I, Kikhia B, Nugent C, Boytsov A, Hallberg J, Synnes K, McClean S, Finlay D. Optimal placement of accelerometers for the detection of everyday activities. *Sensors* 2013;13(7):9183-200.
- [9] Bulling A, Blanke U, Schiele B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys* 2014;46(3):33.
- [10] Fida B, Bernabucci I, Bibbo D, Conforto S, Schmid M. Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer. *Medical Engineering & Physics* 2015; 37(7):705-11.
- [11] Reyes-Ortiz JL, Oneto L, Samà A, Parra X, Anguita D. Transition-aware human activity recognition using smartphones. *Neurocomputing* 2016; 171: 754-67.
- [12] Camci Faith. Change Point Detection in Time Series Data using Support Vector Machines, *International Journal of Pattern Recognition and Artificial Intelligence* 2010;24(1):73-95.
- [13] Downey AB. A Novel Changepoint Detection Algorithm 2008; arXiv preprint arXiv.0812.1237.
- [14] Keogh E, Chu S, Hart D, Pazzani M. An online algorithm for segmenting time series. In: *Proceedings of 1st IEEE International Conference on Data Mining*; 2001 Nov 29- Dec 02; San Jose, California (USA). IEEE; 2001. p 289-96.
- [15] Van Kasteren T; Noulas A; Englebienne G; Kröse B. Accurate activity recognition in a home setting. In: *Proceedings of the 10th International Conference on Ubiquitous computing*; 2008 Sep 21-24; Seoul (South Korea). ACM 2008. p 1-9.
- [16] Achumba IE, Bersch S, Khusainov R, Azzi D, Kamalu U. On time series sensor data segmentation for fall and activity classification. In: *Proceedings of the 14th International Conference on e-Health Networking, Applications and Services*; 2012 Oct 10-13; Beijing (China). IEEE 2012. p 427-30.
- [17] Banos O, Galvez JM, Damas M, Pomares H, Rojas I. Window size impact in human activity recognition. *Sensors* 2014;14(4): 6474-99.
- [18] Kwapisz JR, Weiss GM, Moore SA. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* 2011;12(2):74-82.
- [19] Siirtola P, Rönning J. User-independent human activity recognition using a mobile phone: Offline recognition vs. Real-time on device recognition. *Distributed computing and artificial intelligence* 2012. p 617-27.
- [20] Jiang M, Shang H, Wang Z, Li H, Wang Y. A method to deal with installation errors of wearable accelerometers for human activity recognition. *Physiological Measurement* 2011;32(3):347.
- [21] Chernbumroong S, Cang S, Atkins A, Yu H. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications* 2013;40(5):1662-74.

- [22] Gao L, Bourke AK, Nelson J. Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. *Medical Engineering & Physics* 2014;36(6):779-85.
- [23] Zhang S, Galway L, McClean S, Scotney B, Finlay D, Nugent C. Deriving Relationships between Physiological Change and Activities of Daily Living using Wearable Sensors. *Sensor Systems and Software* 2011; Springer Berlin Heidelberg:235–50.
- [24] Prajapati DR, Mahapatra P B. A new X chart comparable to CUSUM and EWMA charts. *International Journal of Productivity and Quality Management* 2009;4(1):103-28.
- [25] Jain A, Wang Y (n.d.). A New Framework for On-Line Change Detection (Unpublished). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.5929>.
- [26] Patterson T, McClean S, Nugent C, Zhang S, Galway L, Cleland I. Online Change Detection for Timely Solicitation of User Interaction. In: Hervás R, Lee S, Nugent C, Bravo J, editors. *Proceedings of the 8th International Conference on Ubiquitous Computing and Ambient Intelligence*; 2014 Dec 02-05; Belfast (UK). UCAMi; 2014. p 116-23.
- [27] Patterson T, Khan N, McClean S, Nugent C, Zhang S, Cleland I, Qin N. Sensor-Based Change Detection for Timely Solicitation of User Engagement (Accepted subject to revision). *IEEE Transactions on Mobile Computing* 2016.
- [28] Reyes-Ortiz JL, Oneto L, Ghio A, Samá A, Anguita D, Parra X. Human activity recognition on smartphones with awareness of basic activities and postural transitions. In: *Proceedings of the 24th International Conference on Artificial Neural Networks*; 2014 Sep 15-19; Hamburg (Germany). Switzerland: Springer; 2014. P 177-84.
- [29] Atallah L, Lo B, King R, Yang GZ. Sensor positioning for activity recognition using wearable accelerometers. *IEEE Transactions on Biomedical Circuits and Systems* 2011;5(4):320-9.
- [30] Gupta P, Dallas T. Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Transactions on Biomedical Engineering* 2014, 61(6):1780-6.
- [31] Van Lummel R, Ainsworth E, Lindemann U, Zijlstra W, Chiari L, Van Campen P, Hausdorff J. Automated approach for quantifying the repeated sit-to-stand using one body fixed sensor in young and older adults. *Gait & posture* 2013;38(1):153-6.
- [32] Rencher, A. C. *Methods of Multivariate Analysis*. 2nd ed. New York, USA: John Wiley & Sons; 2002.
- [33] Bonferroni. C. E. *Il Calcolo delle Assicurazioni su Gruppi di Teste*. In *Studii in Onore del Profesor S. O. Carboni Roma*; 1936.
- [34] Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine* 2006;10(1):156-67.
- [35] Cheng WC, Jhan DM. Triaxial accelerometer-based fall detection method using a self-constructing cascade-AdaBoost-SVM classifier. *IEEE Journal of Biomedical and Health Informatics* 2013; 17(2):411-9.
- [36] Figo D, Diniz PC, Ferreira DR, Cardoso JM. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing* 2010;14(7):645-62.
- [37] Kang DW, Choi JS, Lee JW, Chung SC, Park SJ, Tack GR. Real-time elderly activity monitoring system based on a tri-axial accelerometer. *Disability and Rehabilitation: Assistive Technology* 2010;5(4):247-53.
- [38] Zhang, M.; Sawchuk, A.A. A feature selection-based framework for human activity recognition using wearable multimodal sensors, In: *Proceedings of the 6th International Conference on Body Area Networks*; 2011 Nov 07-10; Beijing (China). ICST; 2011. p 92-8.
- [39] Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 2005;17(4):491-502.
- [40] Chawla NV. Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*. Springer; 2005. p 853-67.

- [41] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research* 2002;321-57.
- [42] Khoshgoftaar TM, Seiffert C, Hulse JV, Napolitano A, Folleco A. Learning with limited minority class data. In: *Proceedings of the 6th International Conference on Machine Learning and Applications*; 2007 Dec 13-15; Cincinnati (USA). ICMLA; 2007. p 348-53.