

Article

# A User-Adaptive Algorithm for Activity Recognition Based on K-Means Clustering, Local Outlier Factor, and Multivariate Gaussian Distribution

Shizhen Zhao, Wenfeng Li \* and Jingjing Cao

School of Logistics Engineering, Wuhan University of Technology, Wuhan 430070, China; henrylzqlj@whut.edu.cn (S.Z.); bettycao@whut.edu.cn (J.C.)

\* Correspondence: liwf@whut.edu.cn; Tel.: +86-139-8629-2609

Received: 25 April 2018; Accepted: 4 June 2018; Published: 6 June 2018



**Abstract:** Mobile activity recognition is significant to the development of human-centric pervasive applications including elderly care, personalized recommendations, etc. Nevertheless, the distribution of inertial sensor data can be influenced to a great extent by varying users. This means that the performance of an activity recognition classifier trained by one user's dataset will degenerate when transferred to others. In this study, we focus on building a personalized classifier to detect four categories of human activities: light intensity activity, moderate intensity activity, vigorous intensity activity, and fall. In order to solve the problem caused by different distributions of inertial sensor signals, a user-adaptive algorithm based on K-Means clustering, local outlier factor (LOF), and multivariate Gaussian distribution (MGD) is proposed. To automatically cluster and annotate a specific user's activity data, an improved K-Means algorithm with a novel initialization method is designed. By quantifying the samples' informative degree in a labeled individual dataset, the most profitable samples can be selected for activity recognition model adaption. Through experiments, we conclude that our proposed models can adapt to new users with good recognition performance.

**Keywords:** human activity recognition; user-adaptive algorithm; K-Means clustering; local outlier factor; multivariate Gaussian distribution; personalized classifier

## 1. Introduction

Human activity recognition (HAR) is a research field that plays an important role in healthcare systems [1]. It can help people to build a healthy lifestyle with regular physical activities. At present, the methods for HAR are mainly based on computer vision [2] and wearable sensors [3]. The methods based on computer vision use cameras to monitor the movements of a human body and classify the types of human body activities by means of specific algorithms [4,5]. With cameras functioning as the human visual system, these methods use a specific algorithm to simulate the process of human brain judgment. The advantage of camera assistance is that multiple subjects can be detected at the same time without the aid of other devices [6]. Although the accuracy rate is high in some cases, the practical value of this method is undermined by its high requirements of equipment and its tendency to suffer high interference from environmental factors [7]. Moreover, camera-based methods can generate privacy issues [8]. As for the methods based on wearable sensors, with the development of micro-electro-mechanical system (MEMS) technology, human body motion sensors have been miniaturized, thus becoming more flexible and more affordable [9]. Inertial sensor-based HAR can be mainly divided into two steps: feature extraction and classification [9].

Classification techniques commonly used for activity identification based on sensory data were reviewed in [10,11]. Most previous activity recognition research studies created classification models

offline from the training dataset, which contained the data of various people, and transferred these models to different users to validate classification performance. This limits the performance of human activity recognition because the distribution of sensory data is influenced greatly by varying users [12,13]. This is because walking for one user may well be running for another in some cases; therefore, HAR personalization is important for constructing a robust HAR system. In fact, it is hard to generate a user-specific model because of the scarcity of labeled data [14]. Some studies utilized a data generator [15] for data augmentation or designed a framework (e.g., personal digital assistant) [16] for genuine data collection. In contrast, we attempt to design an automatic annotation method for constructing a personalized labeled dataset.

In this paper, we aim to build an adaptive algorithm based on K-Means clustering, local outlier factor (LOF), and multivariate Gaussian distribution (MGD) for activity recognition. We focus on the personalized classifier about its ability to recognize four human activities: light intensity activity (LIA), moderate intensity activity (MIA), vigorous intensity activity (VIA), and fall. The developed algorithm deals with user-specific sensory data to learn and recognize personalized users' activities. Manual data labeling for each user is usually impractical, inconvenient, and time-consuming. To automatically annotate a specific user's dataset, a novel initialization method of a K-Means algorithm is presented. The core idea of this method is to set initial centroids of unclustered dataset as K centroids (with labels) of pre-collected activity dataset. Then, a method based on LOF is proposed to select high confidence samples for personalizing three MGD models. The three MGD models can estimate the probabilities of real-time human activity states including LIA, MIA, and VIA. To further recognize falls, thresholds are set to distinguish a fall from VIA.

The rest of this paper is organized as follows: in Section 2, we present some relative studies that work on physical activity intensity recognition, fall detection, and user-adaptive recognition models without manual interruption. In Section 3, we describe our adaptive algorithm for activity recognition. In Section 4, we set up the experiments and analyze the results. We conclude this paper in Section 5.

## 2. Related Works

In this section, an overview of three main topics related to this paper is given: (1) physical activity intensity recognition; (2) fall detection; and (3) user-adaptive recognition models without manual interruption.

### 2.1. Physical Activity Intensity Recognition

Liu et al. [17] proposed the Accumulated Activity Effective Index Reminder (AAEIReminder), a fuzzy logic prompting mechanism, to help users to manage physical activity. AAEIReminder can detect activity levels using a C4.5 decision tree algorithm and infer activity situations (e.g., the amount of physical activity, days spent exercising) through fuzzy logic. Then it decides whether the situation requires a prompt and how much exercise should be prompted. Liu et al. [18] assessed physical activity and the corresponding energy expenditure through multisensory data fusion (i.e., acceleration and ventilation) based on SVMs. Their experimental results showed that the proposed method is more effective in physical activity assessment and detecting energy expenditure compared with single accelerometer-based methods. Jung et al. [19] used fuzzy logic and an SVM algorithm to enable the classification of datasets from bio-sensors. Then, a decision tree and random forest algorithm was utilized to identify the mental stress level. Finally, a novel activity assessment model based on an expectation maximization (EM) algorithm was proposed to assess users' mental wellness so as to recommend them proper activity. Fahim et al. [20] classified acceleration data from smartphones, using the non-parametric nearest neighbor algorithm, to analyze sedentary lifestyles. The classification process was conducted on the cloud platform, which facilitates users to monitor their long-term sedentary behavior. Ma et al. [21] designed a novel activity level assessment approach for people who suffer from sedentary lifestyles. With their method, users' postures and body swings can be detected using a J48 decision tree algorithm and sensory data from a smart cushion. Then a method based on

the estimation of activity assessment index (AAI) is used to further recognize the activity levels. The aforementioned methods utilize a generic classifier without personalization to assess physical activity. Thus, their performance is limited because the distribution sensory data vary among different people, which leads to inevitable cross-person error.

## 2.2. Fall Detection

Tong et al. [22] proposed a method based on the hidden Markov model (HMM), using tri-axial accelerations, to detect falls. They used the acceleration time series of fall processes before the collision to train the HMM model. Thus, their trained model can not only detect falls but also evaluate fall risks. Bourke et al. [23] described a threshold-based algorithm for fall detection, using the bi-axial angular velocity. Three thresholds for resultant angular velocity, resultant angular acceleration, and resultant angle change were set to distinguish falls from activities of daily living. Sucerquia et al. [24] presented a novel fall detection approach based on a Kalman filter and a non-linear classification feature, using data from a tri-axial accelerometer. Their methodology required a low sampling frequency of only 25 Hz. The experimental results showed that their proposed method has low computational complexity and is robust among embedded systems. Khojasteh et al. [25] compared the performance of threshold-based algorithms and various machine learning algorithms in detecting falls, using data from waist-located tri-axial accelerometers. The experimental results showed that the machine learning algorithms outperformed the threshold-based algorithms. Moreover, among the selected machine learning algorithms, support vector machines provide the highest combination of sensitivity and specificity. Mao et al. [26] re-identified an acceleration threshold of 2.3 g and verified the best sensor location (i.e., waist) of the human body for fall detection. Shi et al. [27] uses J48 decision tree, which is an efficient algorithm derived from C4.5 decision tree [28], to detect falls. The aforementioned methods [22–27] do not consider the effect of personalization on fall detection. Medrano et al. [8] evaluated four algorithms (nearest neighbor (NN), local outlier factor (LOC), one-class support vector machine (One-Class SVM), and SVM) after personalization used as fall detectors to boost their performance when compared to their non-personalized versions. The experimental results showed that there is a general trend towards an increase in performance by detector personalization, but the effect depends on the individual being considered. However, manual labeling is needed in its personalization process, which is impractical in real applications.

## 2.3. User-Adaptive Recognition Model without Manual Interruption

For HAR personalization, manual data labeling for each user is usually impractical, inconvenient, and time-consuming. Some studies automatically personalized their HAR model without human intervention. Viet et al. [29] combined an SVM classifier with K-medoids clustering to build a personalized activity recognition model. Moreover, each user can update the model using his new activities independently. However, training an SVM model requires intensive computation which may not be allowed in a lightweight embedded device. Zhao et al. [30] presented a user-adaptive HAR model adaptation by combining a K-Means clustering algorithm with a decision tree. The outputs of the trained decision tree are organized by the K-Means algorithm. Then, the decision tree is re-trained by the dataset for HAR personalization. Deng et al. [31] presented a fast and accurate cross-person HAR algorithm, which is known as Transfer learning Reduced Kernel Extreme Learning Machine (TransRKELM). Reduced Kernel Extreme Learning Machine (RKELM) is used to build an initial activity recognition model. Then, Online Sequential-Reduced Kernel Extreme Learning Machine (OS-RKELM) is applied to reconstruct the previous model for personalization. Wen et al. [32] utilized AdaBoost to choose the most profitable features automatically during the adaptation process. The initial model is trained by a pre-collected dataset. Dynamically available data sources are then used to adapt and refine the model. Fallahzadeh et al. [33] proposed a cross-person algorithm called Knowledge Fusion-Based Cross-Subject Transfer Learning. First, a source dataset is used to construct an initial activity recognition model (i.e., source model). Second, the source model is tested on a target dataset,

which is collected from a new user, to assign supervised label predictions. Third, the highly similar samples are annotated as a particular activity category if their degree of correlation is higher than a threshold set by a greedy algorithm. Then, the samples of the target dataset are annotated depending on the information acquired from both source and target views to build up an individual dataset. Lastly, an activity recognition model is trained by the personalized labeled dataset. Siirtola et al. [34] used the Learn++ algorithm, which can utilize any classifier as a base classifier, to personalize human activity recognition models. They compared three different base classifiers: quadratic discriminant analysis (QDA), classification and regression tree (CART), and linear discriminant analysis (LDA). The experiment results showed that even a small personalized dataset can improve the classification accuracy, with QDA by 2.0%, CART by 2.3%, and LDA by 4.6%.

Our method is different from those of the abovementioned studies in numerous aspects. Instead of utilizing a generic model to classify or annotate new users' sensory data, we propose a method based on a clustering algorithm to cluster and annotate instances from a new user automatically. Existing HAR adaption methods rely on a trained model that can be updated and adapted to new users, while we consider training a model completely based on new users' sensory data without others' information. Previous activity recognition models usually select high confidence samples for personalization, while we propose a method based on relative density to remove low confidence samples in each data cluster.

### 3. User-Adaptive Algorithm for Activity Recognition

The aim of this research is to design a user-adaptive algorithm for recognizing four categories of human activities: LIA, MIA, VIA, and fall. The process of establishing our proposed method, as shown in Figure 1, includes data collection, data preprocessing, feature extraction and normalization, automatic annotation (K-Means clustering), high confidence samples selection, and model classification. Each step is detailed in the following subsections.

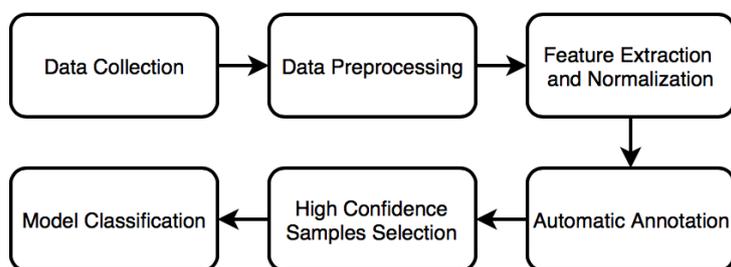


Figure 1. The process of establishing our proposed user-adaptive algorithm.

#### 3.1. Data Preprocessing and Feature Extraction

Acceleration and angular velocity signals [35–37] of the human body can describe states of human activity. Acceleration signals are separated by a Butterworth low-pass filter into gravity and body acceleration. The filter with a 0.3-Hz cutoff frequency is utilized, because the gravitational force has only low frequency components [31]. The data flow is segmented into small windows to deal with large amounts of data, thus facilitating the study and analysis. The size of the sliding window  $T$  and the sampling frequency  $f$  in our study are set to 1 second and 50 Hz, respectively. Therefore, the  $k$ th data units are:  $X_k = \{x_n, x_{n+1}, x_{n+2}, \dots, x_{n+48}, x_{n+49}\}$ ,  $1 \leq k \leq M$ , where  $M$  is the total number of data windows. Through extensive experiments, a fall process was established to take about 300 ms [38], which starts from the beginning of losing balance and continues to the collision of body with lower objects. In order not to cut off the data of the fall process, each data window overlaps with the previous window by 50%. Thus, we set  $n = 25 \times (k - 1)$ . Each sampling point of collected raw

data is:  $x_r = (a_{rx}, a_{ry}, a_{rz}, \omega_{rx}, \omega_{ry}, \omega_{rz})$ . According to the Equation (1), we preprocess the raw data and obtain every sample point:  $x = (a_x, a_y, a_z, \omega_x, \omega_y, \omega_z)$ :

$$\begin{cases} a_x = a_{rx}/k_{ax} + b_{ax} \\ a_y = a_{ry}/k_{ay} + b_{ay} \\ a_z = a_{rz}/k_{az} + b_{az} \\ \omega_x = \omega_{rx}/k_{\omega x} + b_{\omega x} \\ \omega_y = \omega_{ry}/k_{\omega y} + b_{\omega y} \\ \omega_z = \omega_{rz}/k_{\omega z} + b_{\omega z} \end{cases} \quad (1)$$

$k_{ax}, k_{ay}, k_{az}, k_{\omega x}, k_{\omega y}, k_{\omega z}$  are the sensitivity coefficients of three axes. Sensitivity coefficients measure the degree of change of an accelerometer or gyroscope in response to unit acceleration or unit angular velocity changes.  $b_{ax}, b_{ay}, b_{az}, b_{\omega x}, b_{\omega y}, b_{\omega z}$  are their zero drift values. Because the zero drift values are small, their impact is negligible and they were thus ignored in this study.

The quality of feature extraction determines the upper limit of classification performance, and good features can facilitate the process of subsequent classification. Generally, the methods of extracting features from an inertial sensor signal fall into the following three categories: time domain analysis, frequency domain analysis, and time-frequency analysis. Among them, the time domain features are the most commonly used, followed by the frequency domain features. In this study, only the time domain features are extracted because of their lower computational complexity compared with that of the frequency domain feature and the time-frequency feature. The magnitude of synthesized acceleration and angular velocity can be expressed as:  $a = \sqrt{a_x^2 + a_y^2 + a_z^2}$  and  $\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$ . From each window, a vector of 13 features is obtained by calculating variables in the time domain. The mean, standard deviation, energy, mean-crossing rate, maximum value, and minimum value are extracted from the magnitude of synthesized acceleration and angular velocity. In addition, one extra feature, termed tilt angle (TA), can be extracted using Equation (2):

$$TA = \sqrt{\left(\int \omega_x dt\right)^2 + \left(\int \omega_z dt\right)^2} \quad (2)$$

### 3.2. Automatic Annotation Using K-Means Algorithm

In this study, a K-Means algorithm [39] is used to cluster and annotate users' data. The K-Means algorithm has many advantages such as small computational complexity, high efficiency for large datasets, and high linearity of time complexity. However, its clustering results and iteration times are up to the initial cluster centers and the algorithm can function very slowly to converge with a bad initialization [30]. In our solution, a dataset with great similarity to the unclustered dataset is pre-collected. Then the initial center points (with labels) are set to be the cluster centers of this dataset before conducting K-Means clustering steps on the unclustered dataset. After clustering step iterations, the clusters are annotated automatically because of the labeled initial points. The limitation of this method concerns whether a similar dataset can be collected. In this study, the similar dataset can be obtained by experiments.

Let the unclustered dataset  $X = \{x_i | i = 1, \dots, M\}$  be a dataset having K clusters, let  $C = \{c_k | k = 1, \dots, K\}$  be a set of K cluster centers, and let  $S_k = \{x_j^{S_k} | j = 1, \dots, m^{S_k}\}$  be a set of samples that belong to the kth cluster. In order to find the initial centroids that are close to the optimal result, the data of the K clusters are collected previously through experiments. Let  $X^p = \{x_i^p | i = 1, \dots, N\}$  be a pre-collected dataset, let  $C^p = \{c_k^p | k = 1, \dots, K\}$  be a set of K cluster centers of this dataset, and let  $S_k^p = \{x_j^{S_k^p} | j = 1, \dots, n^{S_k^p}\}$  be a set of samples that belong to the kth cluster. The steps of the K-Means clustering we propose are summarized as follows:

- (1) Initialization step: Calculate the  $c_k$  centers as:

$$c_k^0 = c_k^p = \frac{\sum x_j^{S_k^p}}{|S_k^p|} \quad (3)$$

where  $|S_k^p|$  is the number of samples in the  $k$ th cluster of a pre-collected dataset.

- (2) Assignment step: Determine the category of the patterns in one of the  $K$  clusters whose mean has the least squared Euclidean distance.

$$S_k^{(t)} = \left\{ x_i : \|x_i - c_k^{(t)}\|^2 \leq \|x_i - c_p^{(t)}\|^2 \forall j, 1 \leq p \leq K \right\} \quad (4)$$

where each  $x_i$  is assigned to exactly one  $S_k^{(t)}$ .

- (3) Update step: Calculate the new  $c_k^{(t+1)}$  as:

$$c_k^{(t+1)} = \frac{\sum x_j^{S_k^{(t)}}}{|S_k^{(t)}|} \quad (5)$$

$|S_k^{(t)}|$  is the number of samples in the  $k$ th cluster of the unclustered dataset.

- (4) Repeat steps 2 and 3 until there is no change in the assignment step.

The main difference between this K-Means algorithm and the traditional one is that the initial centers we used are labeled and, due to the similarity of different people's activity data, they are close to the expected optimal results.

### 3.3. High Confidence Samples Selection

After conducting automatic annotation on a new user's dataset, confident samples should be selected for MGD model training. Most previous studies [29–31] only selected a certain number of samples in each cluster or chose the samples with high classification confidence. However, these methods ignored the effect of relative density on high confidence samples selection. In our study, some outliers with low relative density are removed from each cluster. Researchers have designed algorithms that take the relative density of a given data instance into consideration to compute the outlier score. LOF [40], a top-notch technique, allows the score to be equal to the average local density ratio of the  $K$ -nearest neighbors of the instance and the local density ratio of the data instance itself. Specifically, the LOF score depends on the values of reachability distance and reachability density. In our study, each sample of a new user has its label after automatic annotation. Also, LOF is used to remove the outliers of each cluster. Accordingly, the  $K$ -nearest neighbors of a sample only include samples in its belonging cluster. Because LOF utilizes the labels of the dataset in our study, we term it label-based local outlier factor (LLOF). If  $d_{kNNbc}(C)$  is the distance to the  $K$ -nearest neighbor in its belonging cluster, then the reachability distance between two samples is defined as follows:

$$d_{reach}(C, D) = \max(d_{kNNbc}(C), d(C, D)) \quad (6)$$

where  $d(C, D)$  represents the Euclidean distance between samples  $C$  and  $D$ .

Taking the maximum in the definition of the reachability distance reduces the statistical fluctuations of  $d(C, D)$  for records that are close to  $C$ . The reachability density is defined as follows:

$$\rho_{reach}(C) = \frac{|N_{kbc}(C)|}{\sum_{D \in N_{kbc}(C)} d_{reach}} \quad (7)$$

where  $N_{kbc}(C)$  is the K-neighborhood of record C, meaning the set of its K-nearest neighbors in its belonging cluster, and  $|N_k(C)|$  is its number. Then, the LLOF score is defined as follows:

$$LLOF(C) = \frac{\sum_{D \in N_{kbc}(C)} \frac{\rho_{reach}(D)}{\rho_{reach}(C)}}{|N_{kbc}(C)|} \quad (8)$$

The LOF score is expected to be close to 1 inside a tight cluster, while it increases for outliers [40]. A threshold  $\varepsilon_1$  is set to filter the outliers.

### 3.4. Training Phase

For the classification technique selection, the rarity of occurrences of falls are an important factor under consideration. Falls occur infrequently and diversely, leading to a lack of related data for training the classifiers [41]. Alternatively, artificial fall data can be collected in controlled laboratory settings, but they may not be the best representatives of actual falls [42]. Moreover, classification models built with artificial falls are more likely to suffer from the problem of over-fitting, caused by time series dataset imbalance [43,44], and may poorly generalize actual falls. In this case, we propose an MGD-based classifier, which does not require fall data in training phase, for detecting LIA, MIA, VIA, and fall. Because it does not require high computational complexity (in the case of low dimensions) and only a few parameters need to be computed in its training phase, it can easily be implemented in wearable embedded systems.

Let  $S_k = \{x_j^{S_k} | j = 1, \dots, m^k\}$  be the kth cluster in the training set X and let each sample  $x_j^{S_k}$  be an dimensional feature vector:  $x_j^{S_k} = (x_1^{S_k}, x_2^{S_k}, \dots, x_n^{S_k})^T$ . The training phase of MGD using  $S_k$  is summarized as follows:

$$\mu_k = \frac{1}{m^k} \sum_{j=1}^{m^k} x_j^{S_k} \quad (9)$$

$$\Sigma_k = \frac{1}{m^k} \sum_{j=1}^{m^k} (x_j^{S_k} - \mu_k)(x_j^{S_k} - \mu_k)^T \quad (10)$$

Given a new example  $x$ ,  $p(x)$  is computed as:

$$p_k(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right) \quad (11)$$

The output  $p(x)$  can be used to estimate the probability that a new example belongs to the same class of  $S_k$ . The example  $x$  is determined to be an anomaly if  $p(x) < \varepsilon$  ( $\varepsilon$  is a threshold value).

Assume that we have a pre-collected dataset  $X^p$  that is similar to an unlabeled dataset X in distribution, and that both datasets include the same data cluster: LIA, MIA, and VIA. The two datasets are collected from different subjects. Since one MGD model only fits one specific class, three MGD models should be trained in order to identify the class of new samples into LIA, MIA, and VIA. When there is a new sample, we assign the class label of maximum of  $p_i(x)$ . Through extensive experiments, falls are classified into VIA in this stage. Then, thresholds are set to distinguish fall from VIA, which are introduced in Section 3.5. The steps of the training phase of our proposed classifier are summarized as follows (Algorithm 1):

**Algorithm 1 Training phase**

**Input:** raw dataset without annotation  $Y = \{y_i | i = 1, \dots, M\}$ , prelabeled dataset

$X^p = \{x_i^p | i = 1, \dots, N\}$ ,  $C^p = \{c_k^p | k = 1, \dots, K\}$ ,  $S_k^p = \{x_j^{S_k^p} | j = 1, \dots, n^{S_k^p}\}$ , max iterations  $T$ , nearest neighbor number  $k$ , outlier threshold  $\varepsilon_1$

**Output:** personalized MGD models  $p_1(x)$ ,  $p_2(x)$ , and  $p_3(x)$

1. Preprocess the dataset  $Y = \{y_i | i = 1, \dots, M\}$  through the Butterworth low-pass filter and dividing by sensitivity coefficient  $k$ .
2. Segment the data into overlapping windows.
3. Extract the features and normalization.
4. Save the results from step 1 and step 2 as  $X = \{x_i | i = 1, \dots, M\}$ .
5. Initialize K-Means clustering centroids  $c_k^0$  by Equation (3).
6. For  $t = 1$  to  $T$  do the following:
  7. Assign samples that belong to the  $k$ th cluster  $S_k$  by Equation (4).
  8. Update  $k$ th centroids  $c_k^{(t+1)}$  by Equation (5).
  9. If there is no change in the assignment step
    10. break
  11. end if
12. end for
13. For  $i = 1$  to  $M$  do the following:
  14.  $score_i = LLOF(x_i)$ ,  $x_i \in X$
  15. if  $score_i > \varepsilon_1$ 
    16. remove  $x_i$  from  $X$
  17. end if
18. end for
19. Using Equation (9) and (10), compute  $\mu_1, \Sigma_1$  by the LIA cluster  $S_1 = \{x_j^{S_1} | j = 1, \dots, m^{S_1}\}$ , compute  $\mu_2, \Sigma_2$  by the MIA cluster  $S_2 = \{x_j^{S_2} | j = 1, \dots, m^{S_2}\}$ , and compute  $\mu_3, \Sigma_3$  by the VIA cluster  $S_3 = \{x_j^{S_3} | j = 1, \dots, m^{S_3}\}$ .
20. Establish the personalized MGD models:  $p_1(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_1|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1)\right)$ ,  
 $p_2(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_2|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_2)^T \Sigma_2^{-1}(x - \mu_2)\right)$ , and  
 $p_3(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_3|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_3)^T \Sigma_3^{-1}(x - \mu_3)\right)$

**3.5. Testing Phase**

The function of our proposed algorithm is to classify human activities into four categories: LIA, MIA, VIA, and fall. VIA and fall can be attributed to the same category due to its large signal variance in a sampling period. Then, the activity state can be determined using Equation (12) and two thresholds,  $\varepsilon_1$  (threshold of  $p_3(x)$ ) and  $\varepsilon_2$  (threshold of  $TA$ ):

$$i' = \underset{i}{\operatorname{argmax}} p_i(x) \quad (12)$$

The  $\varepsilon_2$  is employed to reassure the tilt angle change in a fall movement. If  $i' = 1$ , the real-time activity is determined to be LIA; if  $i' = 2$ , the real-time activity is determined to be MIA; if  $i' = 3$  &  $p_3(x) > \varepsilon_2$  &  $TA < \varepsilon_3$ , the real-time activity is determined to be VIA; if  $i' = 3$  &  $p_3(x) \leq \varepsilon_2$  &  $TA \geq \varepsilon_3$ , the real-time activity is determined to be fall. The overall algorithm is summarized as follows (Algorithm 2):

---

**Algorithm 2 Classifier**


---

**Input:** real-time data A**Output:** human activity category

1. Preprocess the input data through the Butterworth low-pass filter and dividing by sensitivity coefficient  $k$ .
  2. Segment the data into overlapping windows.
  3. Extract the features and construct the feature vector  $x$  with normalization.
  4. Input  $x$  into three MGD models and calculate  $p_1(x)$ ,  $p_2(x)$ , and  $p_3(x)$ .
  5. if  $p_1(x) == \max_{i=1,2,3} p_i(x)$ : Output (LIA)
  6. else if  $p_2(x) == \max_{i=1,2,3} p_i(x)$ : Output (MIA)
  7. else if  $p_3(x) == \max_{i=1,2,3} p_i(x) \ \& \ p_3(x) > \varepsilon_2 \ \& \ TA < \varepsilon_3$ : Output (VIA)
  8. else: Output (Fall)
- 

#### 4. Experimental Section

In this section, we validate our aforementioned methods. We start by introducing our experiment protocol, and then we specify the method for the experimental approach evaluation.

We compare the recognition performance in terms of  $F1$ -measure, which represents the combination of precision and recall, for all the experiments. They are respectively defined as follows:

$$Precision = \frac{T_p}{T_p + F_p} \quad (13)$$

$$Recall = \frac{T_p}{T_p + F_n} \quad (14)$$

$$F - \text{measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (15)$$

where  $T_n$  (true negatives) and  $T_p$  (true positives) are the correct classifications of negative and positive examples, respectively.  $F_n$  (false negatives) denotes the positive examples incorrectly classified into the negative classes. Inversely,  $F_p$  (false positives) represents the negative examples incorrectly classified into the positive classes. Our proposed algorithm can classify human activities into four categories: LIA, MIA, VIA, and fall.  $F$ -measure is computed for each activity type. For example, the output of the algorithm can be treated as non-fall or fall when evaluating the capability of recognizing falls. In our study, the simulations were executed in a MATLAB 2017 environment, which was run on an ordinary PC with 2.60 GHz CPU and 4 Gb memory space.

##### 4.1. Experiment Protocol

Table 1 offers the summary of different activities and activity categories and reports examples of activities that fall under each specific activity category. The definition principle of LIA, MIA, and VIA categories is shown in the table [45]. The Metabolic Equivalent of Task (MET) [46], a physiological measure gauging the energy cost of physical activities, can be applied to measure physical activities. For example, 1 MET is treated as the Resting Metabolic Rate (RMR) obtained during quiet sitting. MET values of activities range from 0.9 (sleeping) to 23 (running at 22.5 km/h). Specifically, MET values of light intensity activities are in the range of (1, 3), moderate intensity activities are in the range of (3, 6), and vigorous intensity activities are in the range of (6, 10). Assuming that a MET value is greater than 9, the user should be engaging intensely vigorous activity [21]. Additionally, we also list fall as an activity category in this table, divided into four types (forward falling, backward falls, left-lateral falling, right-lateral falling).

**Table 1.** Categorization of different activities.

Activity Category	Description	Activities
Light intensity activity (LIA)	Users perform common daily life activities in light movement condition.	Working at a desk, reading a book, having a conversation
Moderate intensity activity (MIA)	Users perform common daily life activities in moderate movement condition.	Walking, walking downstairs, walking upstairs
vigorous intensity activity (VIA)	Users perform vigorous activities to keep fit.	Running, rope jumping
Fall	Users accidentally falls to the ground in a short time.	Forward falling, backward falling, left-lateral falling, right-lateral falling

The conduction of the experiments lasted two weeks with our campus chosen as its location; 10 people (five males and five females) were randomly selected as participants. They were all informed of the purpose and procedure of the study. Their ages were in the range of (20, 45) years, and none of the participants presented an unhealthy status or limb injury. Their BMI measurements were in the range of (16, 34), and the BMI distribution in our experiment sample can be seen in Table 2.

**Table 2.** BMI distribution of the participants.

Description	Underweight	Normal	Overweight and Obese
BMI	<18.5	(18.5, 25)	$\geq 25$
Number of subjects	3	5	2

During the experiment, the participants could choose the time period for each session freely. MPU6050 was used to collect the tri-axial acceleration and the tri-axial angular velocity with sampling rate of 50 Hz. As the waist is the geometric center of the human body [38], the sensor was attached to each person's waist. The  $x$ -axis,  $y$ -axis, and  $z$ -axis of the sensor corresponded to the horizontal (transverse rotation), median (axial rotation), and lateral (sagittal rotation) movements of the human body, respectively. Participants were asked to perform the following activities: (1) working at a computer at a desk; (2) reading a book; (3) having a conversation (in calm status); (4) walking; (5) walking downstairs; (6) walking upstairs; (7) running; (8) rope jumping; (9) forward falling; (10) backward falling; (11) left-lateral falling; and (12) right-lateral falling. On average, experiment sessions (activities 1–8) lasted about 30 min and each participant had to perform 10 sessions over the two weeks. Each participant was asked to simulate each type of genuine fall 60 times on a safety cushion in order to validate the fall detection performance of the MGD-based classifier. In reality, each activity category contains more than the above examples. In our study, we used only one activity type in each category (LIA, MIA, and VIA) to train the MGDs, and used other activities to verify that the classifier can accurately classify the unknown activity. The ratio of the dataset for training and testing was 4:1. The datasets of activity 1, activity 4, and activity 7 were selected to train  $p_1(x)$ ,  $p_2(x)$ , and  $p_3(x)$ . In practice, users were asked to collect their own dataset for model adaption. The selected activities are the commonest activities among their belonging classes, which are more convenient for users to collect the data. The testing dataset included all of the activities in Table 1.

#### 4.2. Classification Performance of Our Proposed Method

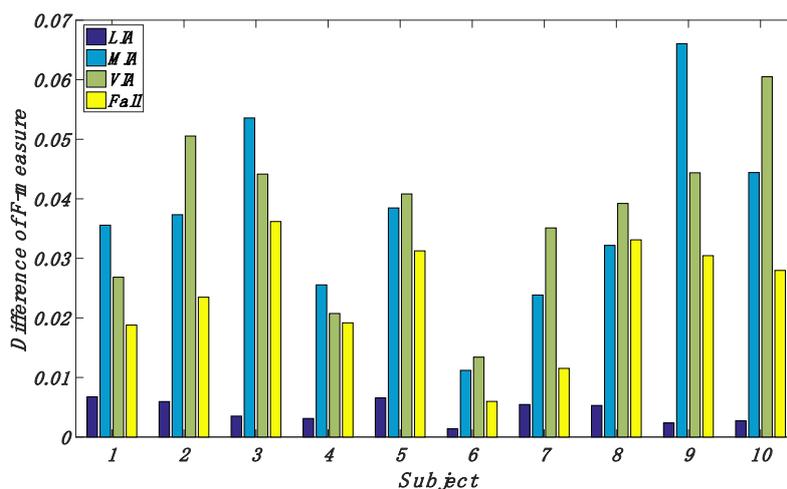
In this subsection, we firstly validate the performance of our method in recognizing LIA, MIA, VIA, and fall. Secondly, in order to evaluate the effect of adaption, the generic model is also trained. To train the personalized classifier, all steps of our algorithm are conducted. Both the training dataset

and the testing dataset include only activity data from the selected subject. The data from the rest of the subjects are used to initialize the centroids in the automatic annotation step. The personalized classifier is also trained for all volunteers. On the part of the generic classifier, leave-one-out (LOO) cross-validation is utilized in training and testing the classifier. Thus, one participant's dataset is left out and the parameters in the MGD models are defined based on the data from the others' datasets. The trained classifier is then tested on the participant's dataset that was left out earlier. The same procedure is repeated for the rest of participants. It is worth mentioning that the collected samples of fall were only included in the test dataset, because the proposed MGD classifier does not require fall data in its training phase. The performance evaluation for our proposed algorithm presented in Table 3 shows that 100% of the cells contain values higher than 0.95 and 38% contain values higher than 0.98. The result shows the best performance to be for LIA, with an F-measure of 0.9875 averaged for all participants. For MIA, VIA, and fall, the mean values of the F-measure reached 0.9729, 0.9692, and 0.9766, respectively.

**Table 3.** F-measure of four recognized activity categories for our proposed algorithm. The last row presents the average values for all participants.

Users	LIA	MIA	VIA	Fall
1	0.9822	0.9766	0.9765	0.9709
2	0.9985	0.9821	0.9673	0.9635
3	0.9963	0.9782	0.9775	0.9836
4	0.9822	0.9673	0.9661	0.9750
5	0.9866	0.9687	0.9642	0.9809
6	0.9782	0.9821	0.9739	0.9774
7	0.9865	0.9680	0.9641	0.9887
8	0.9733	0.9753	0.9661	0.9850
9	0.9978	0.9653	0.9523	0.9711
10	0.9931	0.9651	0.9839	0.9723
<b>Mean</b>	<b>0.9875</b>	<b>0.9729</b>	<b>0.9692</b>	<b>0.9766</b>

In Figure 2, the effect of personalization is shown. The difference is in favor of the personalized classifier for all participants. For nine of the participants, the difference is statistically significant. As shown in Figure 1, the improvement of performance for MIA, VIA, and fall is much more than that for LIA. The estimated averages of the improvement in F-measure are 0.0043, 0.0368, 0.0376, and 0.0238 for the four categories of activity recognition. The one exception's BMI value is between 18.5 and 25. Moreover, we notice that the subjects whose BMI value is larger than 25 or less than 18.5 always have larger difference and we want to test whether any difference of proportions that we observe is significant.  $u_i (i = 1, \dots, 10)$  is set to be the average value of the improvement in F-measure for the  $i$ th subject.  $u_{mean}$  is the average value of  $u_i$  over all subjects. If the subjects' BMI values are between 18.5 and 25, these subjects belong to the Normal group and the rest of them belong to the Abnormal group. If the subjects'  $u_i$  values are greater than  $u_{mean}$ , they are termed significant subjects (SS). Otherwise, they are termed insignificant subjects (IS). Then we assume the null hypothesis that the Normal group and the Abnormal group are equally likely to be SS. The distribution of SS and IS among the Normal group and the Abnormal group is summarized in Table 4. Through Fisher's exact test [47], we can get  $p$ -value  $< 0.05$ , which evidently indicates that Normal group and Abnormal group are not equally to be SS.

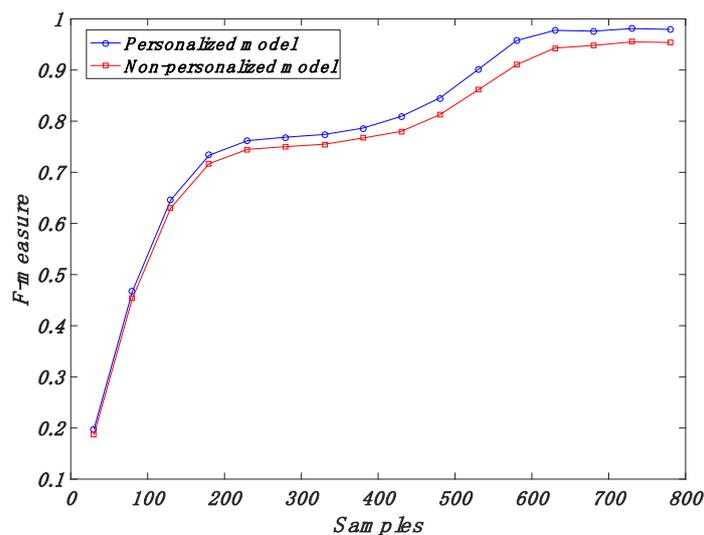


**Figure 2.** Difference in F-measure between a personalized classifier and a generic classifier for four recognized activity categories.

**Table 4.** Distribution of SS and IS among Normal group and Abnormal group.

	Normal	Abnormal	Row total
SS	1	5	6
IS	4	0	4
Column total	5	5	10

In Figure 3, we compare the personalized model and the non-personalized model for different training dataset sizes. The curves show average F-measure values for each defined personal training dataset size and non-personal training dataset size. The average F-measure values for the personalized classifier converged at a higher average F-measure value of around 0.9796, while the non-personalized classifier only reached an F-measure value of around 0.9540.



**Figure 3.** Comparison between personalized models and non-personalized models for different training dataset sizes.

### 4.3. Comparison of the Proposed Algorithm with Previous Studies

In this subsection, we compare our method with the personalization algorithms proposed in the [29,31] as well as fall detection methods reported in [22,27]. The methods proposed in [29,31] have similar workflows for building a personalized classifier. Firstly, they train a generic model before transferring to a new user. Secondly, the trained model is used to classify the unlabeled data from the new user. Thirdly, confident samples are selected to reconstruct the previous model. The algorithms and parameter settings for both methods are summarized in Table 5. The simulations for SVM were conducted using LIBSVM [48] packages and J48 is simulated using WEKA toolbox [49]. K-Medoids, TransRKELM, and HMM were implemented by us using Matlab. The Gaussian kernel  $\mathcal{K}(x, x_i) = \exp(-x - x_i^2/\sigma)$  was used in SVM and RKELM in their studies. For the Gaussian kernel parameter  $\sigma$ , the generalized accuracy is estimated by using different parameter values  $\sigma = [2^{-5}, \dots, 2^{20}]$ , and choosing the value with the best performance. For RKELM, the generalized performance is estimated, using various combinations of regularized parameter  $\lambda$  and the subset size  $\tilde{n}$ :  $\lambda = [2^{-5}, \dots, 2^{30}]$ , and  $\tilde{n}$  is increased gradually with an interval of 5. For SVM, the generalized performance was estimated using various combinations of parameters  $\lambda$  and  $K$  (the number of confident samples for each cluster):  $\lambda = [2^{-5}, \dots, 2^{30}]$ , and  $K$  is increased by an interval of 5. In order to compare with the HMM-based fall detection algorithm proposed in Reference [22], the HMM was initialized as follows: the number of invisible states  $M = 3$ , the number of observation values  $N = 8$ , and initial state distribution  $\pi_1 = 1$ ,  $\pi_i = 0 (i = 2, \dots, M)$ . Then, the generated performance was estimated using different thresholds of  $P_2 = [0.1, \dots, 0.2]$ . For J48, the generated performance is estimated by using different combination of the pruning confidence  $C$  and the minimum number of instances per leaf  $M$ :  $C = [0.01, \dots, 1]$  and  $M$  is increased by an interval of 1. Twenty trials of simulations for each aforementioned parameter or parameter combination were carried out. The best performances obtained are shown in this paper.

**Table 5.** Details of two models proposed in literature.

Author	Algorithm	Parameter Setting
Viet et al. [29]	K-Medoids, SVM	$K = 95, \lambda = 2^{15}, \sigma = 2^{10}$
Deng et al. [31]	TransRKELM	$\sigma = 2^{10}, \tilde{n} = 400, \lambda = 2^{25}$
Tong et al. [22]	HMM	$M = 3, N = 8, \pi_1 = 1, \pi_i = 0 (i = 2, \dots, M), P_2 = 0.123$
Shi et al. [27]	J48	$C = 0.25, M = 2$

As can be seen from Table 6, our proposed method outperforms others in recognizing LIA, MIA, and VIA. The performance of fall detection is not the best among them, but its F-measure, 0.9766, is still competitive. Moreover, the fall samples are not required in our training phase. The training time in the table represents the time of adapting a model to a specific user's dataset. In addition, the average training time (20 trials) of our method is 3.56 s, while the other two methods consume 10.12 s and 0.93 s. Moreover, the average testing time (20 trials) of our method is 0.02 s, while the other two methods consume 2.83 s and 0.15 s. This indicates that our method is much faster than the methods presented in [29,31] if embedded in real-time activity recognition applications.

**Table 6.** Performance of the user-adaptive methods.

Author	LIA	MIA	VIA	Fall	Training Time (s)	Testing Time (s)
Viet et al. [29]	0.9844	0.9230	0.9376	0.9845	10.12	2.83
Deng et al. [31]	0.9854	0.9423	0.9364	0.9813	0.93	0.15
Our proposed method	0.9875	0.9729	0.9692	0.9766	3.56	0.02

In Table 7, we compare the fall detection performance of our method with two efficient algorithms used in [22,27]. Our algorithm shows the highest F-measure values of 0.9766. The J48 algorithm [27] has the least training time of 2.03 s, but its F-measure value of 0.9410 is not competitive. Our proposed algorithm is the most efficient in testing phase with testing time of 0.02 s, while the others consume 0.04 s and 0.05 s.

**Table 7.** Performance of the fall detection methods.

Author	Fall	Training Time (s)	Testing Time (s)
Tong et al. [22]	0.9508	4.02	0.04
Shi et al. [27]	0.9410	2.03	0.05
Our proposed method	0.9766	3.56	0.02

## 5. Conclusions

In conclusion, a user-adaptive algorithm for activity recognition based on K-Means clustering, LOF, and MGD is presented. In order to personalize our classifier, we proposed a novel initialization method for a K-Means algorithm that is based on a pre-collected dataset which has been already labeled. In our study, we used a pre-collected activity dataset to generate initial centroids of K-Means clustering. After the iterations, the dataset was clustered and labeled by the centroids. Then, LLOF was used to select high confidence samples for model personalization.

Due to inadequate fall data used for training a supervised learning algorithm, anomaly detection can be employed for falls detection. MGD is an effective classification technique, requiring only a few computational steps either in training phases or activity classification processes. Thus, it has low computational complexity compared with some computation-demanding classifiers. Moreover, it can produce an accurate classification performance with carefully selected features. Three MGD models were trained by a specific user's dataset. LIA, MIA, and VIA could be differentiated through comparing probability values, which were outputs of  $p_1(x)$ ,  $p_2(x)$ , and  $p_3(x)$ .  $\varepsilon_2$  and  $\varepsilon_3$  were set to distinguish falls from VIA.

The experimental results showed that our personalized algorithm can effectively detect LIA, MIA, VIA, and falls with F-measures of 0.9875, 0.9729, 0.9692, and 0.9766, respectively. Moreover, compared with the algorithms in the literature [22,27,29,31], our proposed algorithm has lower computational complexity in the testing phase, with a testing time of 0.02 s.

In our future work, we plan to apply multi-sensor fusion techniques [50,51] and deep learning methods to detect more complicated abnormal behaviors, which can be put solid ground for the design and implementation of a real system for real-time physical activity recognition. Additionally, we will engage in testing our proposed method using genuine fall datasets.

**Author Contributions:** S.Z. conceived of and developed the algorithms, performed the experiments, analyzed the results, and drafted the initial manuscript. J.C. gave some valuable suggestions to this paper. W.L. supervised the study and contributed to the overall research planning and assessment.

**Acknowledgments:** This research is financially supported by the China-Italy Science and Technology Cooperation project "Smart Personal Mobility Systems for Human Disabilities in Future Smart Cities" (China-side Project ID: 2015DFG12210), the National Natural Science Foundation of China (Grant No: 61571336 and 61502360). The hardware used in the experiment is supported by Joint Lab of Internet of Things Tech in School of Logistics Engineering, Wuhan University of Technology. The authors would like to thank the volunteers who participated in the experiments for their efforts and time.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Davila, J.C.; Cretu, A.M.; Zaremba, M. Wearable Sensor Data Classification for Human Activity Recognition Based on an Iterative Learning Framework. *Sensors* **2017**, *17*, 1287. [[CrossRef](#)] [[PubMed](#)]

2. Crispim-Junior, C.F.; Gómez Uría, A.; Strumia, C.; Koperski, M.; König, A.; Negin, F.; Cosar, S.; Nghiem, A.T.; Chau, D.P.; Charpiat, G.; et al. Online Recognition of Daily Activities by Color-Depth Sensing and Knowledge Models. *Sensors* **2017**, *17*, 1528. [[CrossRef](#)] [[PubMed](#)]
3. Janidarmian, M.; Roshan Fekr, A.; Radecka, K.; Zilic, Z. A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition. *Sensors* **2017**, *17*, 529. [[CrossRef](#)] [[PubMed](#)]
4. Yang, X.; Tian, Y.L. Super normal vector for human activity recognition with depth cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1028–1039. [[CrossRef](#)] [[PubMed](#)]
5. Li, W.; Wong, Y.; Liu, A.A.; Li, Y.; Su, Y.T.; Kankanhalli, M. Multi-camera action dataset for cross-camera action recognition benchmarking. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 187–196.
6. Chen, C.; Jafari, R.; Kehtarnavaz, N. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools Appl.* **2017**, *76*, 4405–4425. [[CrossRef](#)]
7. Wang, S.; Zhou, G. A review on radio based activity recognition. *Digital Commun. Netw.* **2015**, *1*, 20–29. [[CrossRef](#)]
8. Medrano, C.; Plaza, I.; Igual, R.; Sánchez, Á.; Castro, M. The effect of personalization on smartphone-based fall detectors. *Sensors* **2016**, *16*, 117. [[CrossRef](#)] [[PubMed](#)]
9. Chen, Y.; Xue, Y. A deep learning approach to human activity recognition based on single accelerometer. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kowloon, China, 9–12 October 2015; pp. 1488–1492.
10. Preece, S.J.; Goulermas, J.Y.; Kenney, L.P.; Howard, D.; Meijer, K.; Crompton, R. Activity identification using body-mounted sensors—A review of classification techniques. *Physiol. Meas.* **2009**, *30*, R1. [[CrossRef](#)] [[PubMed](#)]
11. Safi, K.; Attal, F.; Mohammed, S.; Khalil, M.; Amirat, Y. Physical activity recognition using inertial wearable sensors—A review of supervised classification algorithms. In Proceedings of the International Conference on Advances in Biomedical Engineering (ICABME), Beirut, Lebanon, 16–18 September 2015; pp. 313–316.
12. Weiss, G.M.; Lockhart, J.W. *The Impact of Personalization on Smartphone-Based Activity Recognition*; American Association for Artificial Intelligence: Palo Alto, CA, USA, 2012; pp. 98–104.
13. Ling, B.; Intille, S.S. Activity recognition from user-annotated acceleration data. *Proc. Pervasive* **2004**, *3001*, 1–17.
14. Saez, Y.; Baldominos, A.; Isasi, P. A Comparison Study of Classifier Algorithms for Cross-Person Physical Activity Recognition. *Sensors* **2017**, *17*, 66. [[CrossRef](#)] [[PubMed](#)]
15. Okeyo, G.O.; Chen, L.; Wang, H.; Sterritt, R. Time handling for real-time progressive activity recognition. In Proceedings of the 2011 International Workshop on Situation Activity & Goal Awareness, Beijing, China, 18 September 2011; pp. 37–44.
16. Pärkkä, J.; Cluitmans, L.; Ermes, M. Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 1211–1215. [[CrossRef](#)] [[PubMed](#)]
17. Liu, C.T.; Chan, C.T. A Fuzzy Logic Prompting Mechanism Based on Pattern Recognition and Accumulated Activity Effective Index Using a Smartphone Embedded Sensor. *Sensors* **2016**, *16*, 1322. [[CrossRef](#)] [[PubMed](#)]
18. Liu, S.; Gao, R.X.; John, D.; Staudenmayer, J.W.; Freedson, P.S. Multisensor data fusion for physical activity assessment. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 687–696. [[PubMed](#)]
19. Jung, Y.; Yoon, Y.I. Multi-level assessment model for wellness service based on human mental stress level. *Multimed. Tools Appl.* **2017**, *76*, 11305–11317. [[CrossRef](#)]
20. Fahim, M.; Khattak, A.M.; Chow, F.; Shah, B. Tracking the sedentary lifestyle using smartphone: A pilot study. In Proceedings of the International Conference on Advanced Communication Technology, Pyeongchang, Korea, 31 January–3 February 2016; pp. 296–299.
21. Ma, C.; Li, W.; Gravina, R.; Cao, J.; Li, Q.; Fortino, G. Activity level assessment using a smart cushion for people with a sedentary lifestyle. *Sensors* **2017**, *17*, 2269. [[CrossRef](#)] [[PubMed](#)]
22. Tong, L.; Song, Q.; Ge, Y.; Liu, M. Hmm-based human fall detection and prediction method using tri-axial accelerometer. *IEEE Sens. J.* **2013**, *13*, 1849–1856. [[CrossRef](#)]
23. Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys.* **2008**, *30*, 84–90. [[CrossRef](#)] [[PubMed](#)]
24. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. Real-Life/Real-Time Elderly Fall Detection with a Triaxial Accelerometer. *Sensors* **2018**, *18*, 1101. [[CrossRef](#)] [[PubMed](#)]

25. Aziz, O.; Musngi, M.; Park, E.J.; Mori, G.; Robinovitch, S.N. A comparison of accuracy of fall detection algorithms (threshold-based vs. Machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Med. Biol. Eng. Comput.* **2017**, *55*, 45–55. [[CrossRef](#)] [[PubMed](#)]
26. Mao, A.; Ma, X.; He, Y.; Luo, J. Highly Portable, Sensor-Based System for Human Fall Monitoring. *Sensors* **2017**, *17*, 2096. [[CrossRef](#)] [[PubMed](#)]
27. Shi, G.; Zhang, J.; Dong, C.; Han, P.; Jin, Y.; Wang, J. Fall detection system based on inertial mems sensors: Analysis design and realization. In Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, Shenyang, China, 8–12 June 2015; pp. 1834–1839.
28. Zhao, S.; Li, W.; Niu, W.; Gravina, R.; Fortino, G. Recognition of human fall events based on single tri-axial gyroscope. In Proceedings of the 15th IEEE International Conference on Networking, Sensing and Control, Zhuhai, China, 26–29 March 2018; pp. 1–6.
29. Viet, V.Q.; Thang, H.M.; Choi, D. Personalization in Mobile Activity Recognition System Using K-Medoids Clustering Algorithm. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 797–800.
30. Zhao, Z.; Chen, Y.; Liu, J.; Shen, Z.; Liu, M. Cross-people mobile-phone based activity recognition. In Proceedings of the International Joint Conference on Artificial Intelligence, Barcelona, Spain, 19–22 July 2011; pp. 2545–2550.
31. Deng, W.Y.; Zheng, Q.H.; Wang, Z.M. Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Netw.* **2014**, *53*, 1–7. [[CrossRef](#)] [[PubMed](#)]
32. Wen, J.; Wang, Z. Sensor-based adaptive activity recognition with dynamically available sensors. *Neurocomputing* **2016**, *218*, 307–317. [[CrossRef](#)]
33. Fallahzadeh, R.; Ghasemzadeh, H. Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data. In Proceedings of the 8th International Conference on Cyber-Physical Systems, Pittsburgh, PA, USA, 18–20 April 2017; pp. 293–302.
34. Siirtola, P.; Koskimäki, H.; Röning, J. Personalizing human activity recognition models using incremental learning. In Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2018), Brugge, Belgium, 25–27 April 2018; pp. 627–632.
35. Greene, B.R.; Doheny, E.P.; Kenny, R.A.; Caulfield, B. Classification of frailty and falls history using a combination of sensor-based mobility assessments. *Physiol. Meas.* **2014**, *35*, 2053–2066. [[CrossRef](#)] [[PubMed](#)]
36. Pereira, F. Developments and trends on video coding: Is there a xVC virus? In Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 31 January–1 February 2008; pp. 384–389.
37. Giansanti, D.; Maccioni, G.; Cesinaro, S.; Benvenuti, F.; Macellari, V. Assessment of fall-risk by means of a neural network based on parameters assessed by a wearable device during posturography. *Med. Eng. Phys.* **2008**, *30*, 367–372. [[CrossRef](#)] [[PubMed](#)]
38. Huang, C.L.; Chung, C.Y. A real-time model-based human motion tracking and analysis for human-computer interface systems. *Eurasip J. Adv. Sign. Proces.* **2004**, *2004*, 1–15. [[CrossRef](#)]
39. Peters, G. Some refinements of rough K-Means. *Pattern Recognit.* **2006**, *39*, 1481–1491. [[CrossRef](#)]
40. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. Lof: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; pp. 93–104.
41. Khan, S.S.; Hoey, J. Review of fall detection techniques: A data availability perspective. *Med. Eng. Phys.* **2017**, *39*, 12–22. [[CrossRef](#)] [[PubMed](#)]
42. Kangas, M.; Vikman, I.; Nyberg, L.; Korpelainen, R.; Lindblom, J.; Jämsä, T. Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. *Gait Posture* **2012**, *35*, 500–505. [[CrossRef](#)] [[PubMed](#)]
43. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.
44. De la Cal, E.; Villar, J.; Vergara, P.; Sedano, J.; Herrero, A. A smote extension for balancing multivariate epilepsy-related time series datasets. In Proceedings of the 12th International Conference on Soft Computing Models in Industrial and Environmental Applications, León, Spain, 6–8 September 2017; pp. 439–448.
45. Metabolic Equivalent Website. Available online: [https://en.wikipedia.org/wiki/Metabolic\\_equivalent](https://en.wikipedia.org/wiki/Metabolic_equivalent) (accessed on 10 September 2017).

46. Alshurafa, N.; Xu, W.; Liu, J.J.; Huang, M.C.; Mortazavi, B.; Roberts, C.K.; Sarrafzadeh, M. Designing a robust activity recognition framework for health and exergaming using wearable sensors. *IEEE J. Biomed. Health Inf.* **2014**, *18*, 1636. [[CrossRef](#)] [[PubMed](#)]
47. Hess, A.S.; Hess, J.R. Understanding tests of the association of categorical variables: The pearson chi-square test and fisher's exact test. *Transfusion* **2017**, *57*, 877–879. [[CrossRef](#)] [[PubMed](#)]
48. Chang, C.C.; Lin, C.J. Libsvm: A library for support vector machines. *ACM Trans. Intel. Syst. Technol.* **2011**, *2*, 1–27. [[CrossRef](#)]
49. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [[CrossRef](#)]
50. Gravina, R.; Alinia, P.; Ghasemzadeh, H.; Fortino, G. Multi-Sensor Fusion in Body Sensor Networks: State-of-the-art and research challenges. *Inf. Fusion* **2017**, *35*, 68–80. [[CrossRef](#)]
51. Cao, J.; Li, W.; Ma, C.; Tao, Z.; Cao, J.; Li, W. Optimizing multi-sensor deployment via ensemble pruning for wearable activity recognition. *Inf. Fusion* **2017**, *41*, 68–79. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).