sensors

MDPI

Article

# Transition Activity Recognition System Based on Standard Deviation Trend Analysis

**Junhao Shi, Decheng Zuo * and Zhan Zhang**

Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China; shijunhao@hit.edu.cn (J.S.); zz@ftcl.hit.edu.cn (Z.Z.)

* Correspondence: zuodc@ftcl.hit.edu.cn

Received: 27 April 2020; Accepted: 29 May 2020; Published: 31 May 2020

check for updates

**Abstract:** With the development and popularity of micro-electromechanical systems (MEMS) and smartphones, sensor-based human activity recognition (HAR) has been widely applied. Although various kinds of HAR systems have achieved outstanding results, there are still issues to be solved in this field, such as transition activities, which means the transitional process between two different basic activities, discussed in this paper. In this paper, we design an algorithm based on standard deviation trend analysis (STD-TA) for recognizing transition activity. Compared with other methods, which directly take them as basic activities, our method achieves a better overall performance: the accuracy is over 80% on real data.

**Keywords:** human activity recognition; transition activity; smartphone; SVM; trend analysis

## 1. Introduction

Human activity recognition (HAR) is a significant subfield of pervasive computing and provides important context information for an ocean of applications such as medical care, education, and entertainment [1–3]. In recent decades, with the popularity of smart phones and other wearable devices, HAR applications based on built-in inertial sensors have been significantly developed. For example, WooSeok Hyun et al. [4] developed a wireless body sensor network which integrated different physiological sensors to sense physiological data from a human body with smartphones as computing platform. Chetty Girija et al. [5] realized an automatic and intelligent daily activity monitoring application for elderly people using smartphone inertial sensors. Espinilla Macarena et al. [6] designed a subwindow-based online recognition architecture, which achieved promising results in their experiments. Garcia-Ceja et al. [7] used sound and accelerometer data collected with a smartphone and a wristband while performing home task activities, and the whole system performed outstandingly using a multi-view stacking method. Zahin et al. [8] proposed a semi-supervised classifier, mainly using deep learning networks, in sensor-based smart health monitoring. In summary, most researchers pursue higher activity recognition accuracy.

Although most related works achieve excellent results in recognizing daily activities (e.g., walking, sitting, standing et al.) [9,10], there are still issues in HAR systems that affect its performance. One of them is transition activity, which is the transitional process between two different basic activities, as Figure 1 depicts.

Unlike basic daily activities, transition activity is usually transient and accompanied with intense changes. To the best of the authors' knowledge, most related works choose to ignore the existence of these activities. Sometimes it may be an effective measure for specific practical needs. However, this will inevitably affect the performance of the HAR systems. HAR systems are faced with random noise in the real scenario, which obviously affects the performance of the overall system. Transition activity recognition helps to segment different basic activities to reduce the error rate. Moreover, some HAR systems focus on identifying the transitional activities, such as fall detection systems, require accurate recognition accuracy of these kind of activities. Therefore, transition activity recognition is necessary for HAR systems.

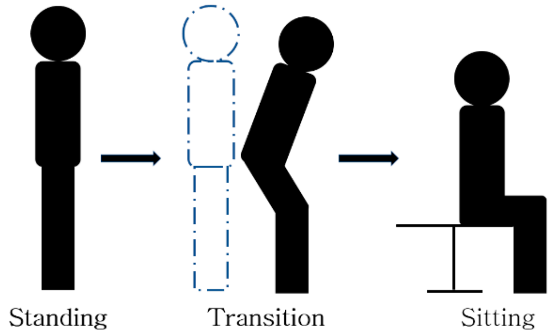



**Figure 1.** Example of transition activity.

Several efforts attempt to solve this issue from their own perspectives. Reyes-Ortiz et al. [11] designed an architecture for recognizing transition activity. They used the probabilistic output of consecutive activity predictions as the main basis for identification and achieved promising results. Mohd et al. [12] took advantage of multivariate Gaussian distribution to judge whether the current activity is a transition activity. The above research mainly focuses on the transition between static activities, while there is relatively little discussion on transitions involving dynamic activities (e.g., standing↔walking).

In this paper, we build a HAR system based on smart phone built-in sensors. The overall work is depicted as follows:

1. We fuse two types of sensor data for accurately recognizing basic daily activities in our dataset.
2. We distinguish the transition activities from basic activities by analyzing the trend of standard deviation.
3. We develop an android application on a smartphone and conduct experiments in a real scenario.

The rest of this paper is organized as follows. In Section 2, we introduce related work, including HAR systems and approaches for recognizing transition activities. Section 3 gives the details of our proposed system architecture and algorithm. Section 4 covers the results of the experiment, and Section 5 is the conclusion and future work.

## 2. Related Works

### 2.1. Human Activity Recognition Systems

With the emergence of Micro-Electromechanical Systems (MEMS), researchers use a variety of professional sensing devices in the HAR systems. Pansiot et al. [13] developed the e-ar sensor, which can be worn on the ear to detect human body signs data for health care. Minnen et al. [14] put multiple sensors on a military suit to recognize tactical actions and provide battlefield information. Moreover, due to the popularity of smart phones, there are also studies developing HAR systems using smartphone built-in sensors. For example, Akhavian et al. [15] bound mobile phones to the upper arm of construction workers to recognize the workers' ongoing activities. Wang et al. [16] developed a system to identify students' behavior through a variety of built-in sensors in smartphones to evaluate their mental health and academic performance. Bisio et al. [17] used smart phones to realize telemedicine monitoring. Ronao et al. [18] proposed a two-stage continuous hidden Markov model (CHMM) approach for the task of activity recognition using accelerometer and gyroscope sensory data gathered from a smartphone. Lu et al. [19] designed an efficient and flexible framework for activity recognition based on smartphone sensors, and the proposed method was independent of device placement and orientation. Most similar HAR systems take advantage of smart devices for collecting sensing data and utilize machine learning algorithms (e.g., Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbor (KNN) [20–22]) for activity recognition. However, they choose to ignore the existence of transition activity instead of taking it as an important issue.

### 2.2. *Transition Activity Recognition*

For some studies, ignoring transition activity causes little influence on the result because they focus on long-term activity monitoring. However, there are also studies that address the transition issue. For example, Song et al. [23] used a 3D accelerometer for activity data collection, and they designed an event-based recognition architecture to deal with transition activity. In the method they utilized a hidden Markov model (HMM), which needed prior probability information, as classifier. Aminikhanghahi et al. [24] built a smart-home environment, placed various sensors on the human body and room doors to collect daily activity data and realized data segmentation and transition activity recognition through change point detection. This research mainly dealt with complex daily activities such as cooking. A similar scenario is also mentioned in Atallah et al. [25]. They used a method based on manifold embedding to map high-dimensional data into low-dimensional space. However, the recognition accuracy on standing-sitting was not high. Moreover, with the development of artificial intelligence, a quantity of works on transition activity regarded transition activity as a new kind of activity similar to the basic ones. Thien Huynh-The et al. [26] used SVM as a classifier to recognize transition activities. The approach was supported by excellent machine learning models. However, the transition activities can be easily classified as other activities in real scenarios, and vice versa.

In this paper, we refer to some ideas from the above works and identify transition activities by analyzing the standard deviation trend. As a simple recognition architecture, the system still achieves promising performance.

## 3. Method and Architecture

### 3.1. *System Architecture*

In this part, we introduce the whole process of activity recognition, as shown in Figure 2.

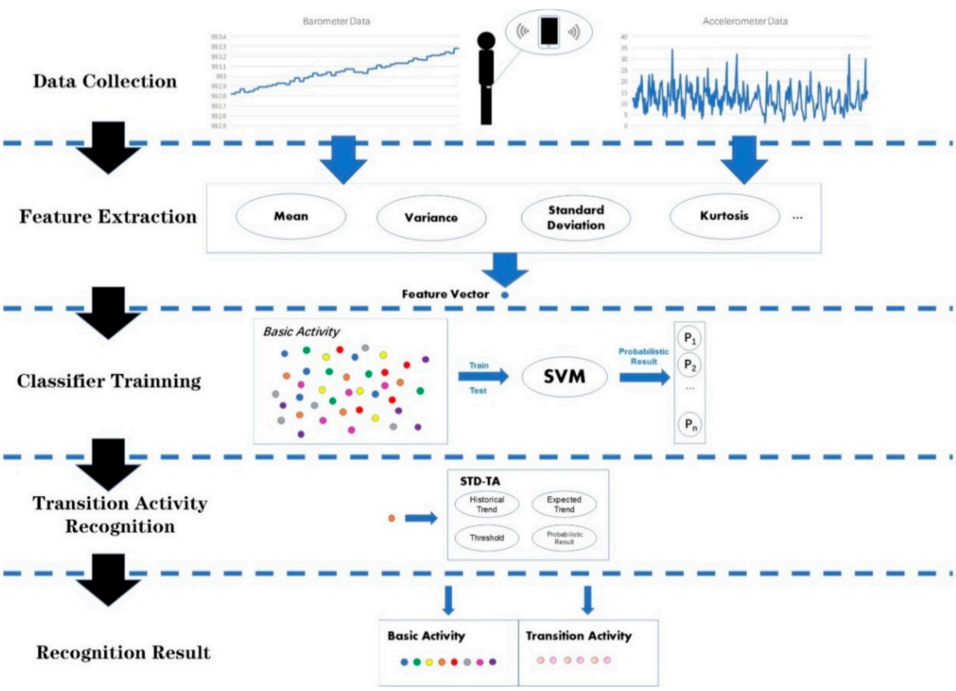



**Figure 2.** Overall activity recognition process.

First, based on the results of our previous work [27], we put a smart phone on the right leg of subjects to collect data in order to achieve the best classification results. The data includes accelerometer data and barometer data. We separate the set into two types: basic activity and transition activity. Then we segment data via a sliding window algorithm and divide all the segments into two parts, one for training and the other for testing. In the feature extraction stage, we extract the statistical features of data segments to reduce the dimension of data and facilitate the training of classifiers. Considering the excellent performance of SVM in pattern recognition [28,29], we train a SVM model as the core classifier in the classifier training stage. SVM outputs the recognition results in the form of probability, which is namely the probabilistic results. In the transition activity recognition stage, we take advantage of the standard deviation trend analysis (STD-TA) method to judge whether the current activity is a transition activity. The final results are achieved in the recognition result stage, and transition activity is distinguished from basic activity.

In the next parts, we give the details of each stage.

### 3.2. Data Collection

We collect eight simple daily activities in our work. Table 1 shows more information about our dataset.

**Table 1.** Information of our dataset.

| | | |
|---|---|---|
| **Basic Activity** | | Sitting, Standing, Lying, Walking, Upstairs, Down Stairs, Running, QuickWalk |
| **Sensors (Sample Frequency)** | | 1 Barometer (5 Hz),<br>1 3D-Accelerometer (50 Hz) |
| **Subjects** | **No. of Subjects** | 10 |
| | **Age Range** | 25–40 |
| | **Male/Female** | 7/3 |
| | **Height Range** | 155 cm~180 cm |
| **Time of Single Collection** | | 3 min |
| **Size of Time Window** | | 1 s (50 data points) |
| **Overlap** | | 0.5 s (25 data points) |

Most activities can be recognized via processing the accelerometer data. According to our previous experience, we set the accelerometer data sampling frequency at 50 Hz, which is enough to achieve outstanding recognition accuracy. We gather the barometer data for recognizing different motion patterns with huge similarity, and 5 Hz is a proper sampling rate given that it changes little in a few milliseconds. For example, the characteristics of upstairs and downstairs in acceleration are highly similar, it is tough to distinguish these two activities unless taking the barometer data into consideration. This is also the general idea of studies on sensor fusion [30]. We apply an upsampling process to make every 10 accelerometer data samples correspond to the same 10 barometer samples to facilitate model training in the following stage.

We collect 3 min data for each activity of each subject (80 s for upstairs and downstairs). For dynamic activities, we ask the subject to perform the same activity for 3 min, and each group of data is equivalent to a periodic repetition of single activity. Based on the actual observation on the data curve, we finally determined the single duration of these activities. For transition activities, we ask each subject to repeat two basic activities within 3 min, and the sequence of activities we obtained might be like A, shown below:

$$A = \{\text{Walking, Transition, Standing, Transition, Walking, Transition, Standing}\}$$

We get about 10 transition activities per collection, which is enough for us to determine the duration of a single transition activity with observation. We make a statistic on the duration of these activities, as shown in Table 2. With the sliding window algorithm, we segment the original data into data segments, which represent single complete activities. The window size is determined by the duration of a single activity. Static activities such as standing and sitting are a long-term static state. We make a statistic on the duration of other activities, as shown in Table 2.

**Table 2.** Duration of each activity.

| Activity | Time (Data Point) |
|---|---|
| downstairs | 50–60 |
| quickwalk | 50–60 |
| upstairs | 50–60 |
| walking | 50–60 |
| running | 30–50 |
| downstairs↔standing | 50 |
| lying↔sitting | 160 |
| quickwalk↔standing | 30 |
| sitting↔standing | 150 |
| upstairs↔standing | 40 |
| standing↔walking | 45 |
| walking↔quickwalk | 10–20 |
| walking↔running | 10–20 |

Compared with basic activities, the proportion of transition activities is relatively low in the real scenario. Therefore, the window size mainly depends on the basic activities, especially the duration of single dynamic activities. As shown in Table 2, the duration of dynamic activities is about 50 data points, and the 1 s time window can well cover all dynamic activities and part of the transition activities. For some tiny short transition activities, such as "Walking to Running", it may fall into the same segment as some other dynamic activities. Given the transitional characteristic of this segment, we consider it as a transition activity. Based on the above information, we finally set the window size as 1 s (50 data sample points).

### *3.3. Feature Extraction & Classifier Training*

Feature extraction can provide the classifier training with statistical characteristics of original data. We get four columns of data after data collection in every data segment, which are named as (Acc_x, Acc_y, Acc_z, Baro). At this stage, we extract statistic features, including mean, variance, standard deviation (STD), etc, of all data segments. More details are given in Table 3.

**Table 3.** Duration of each activity.

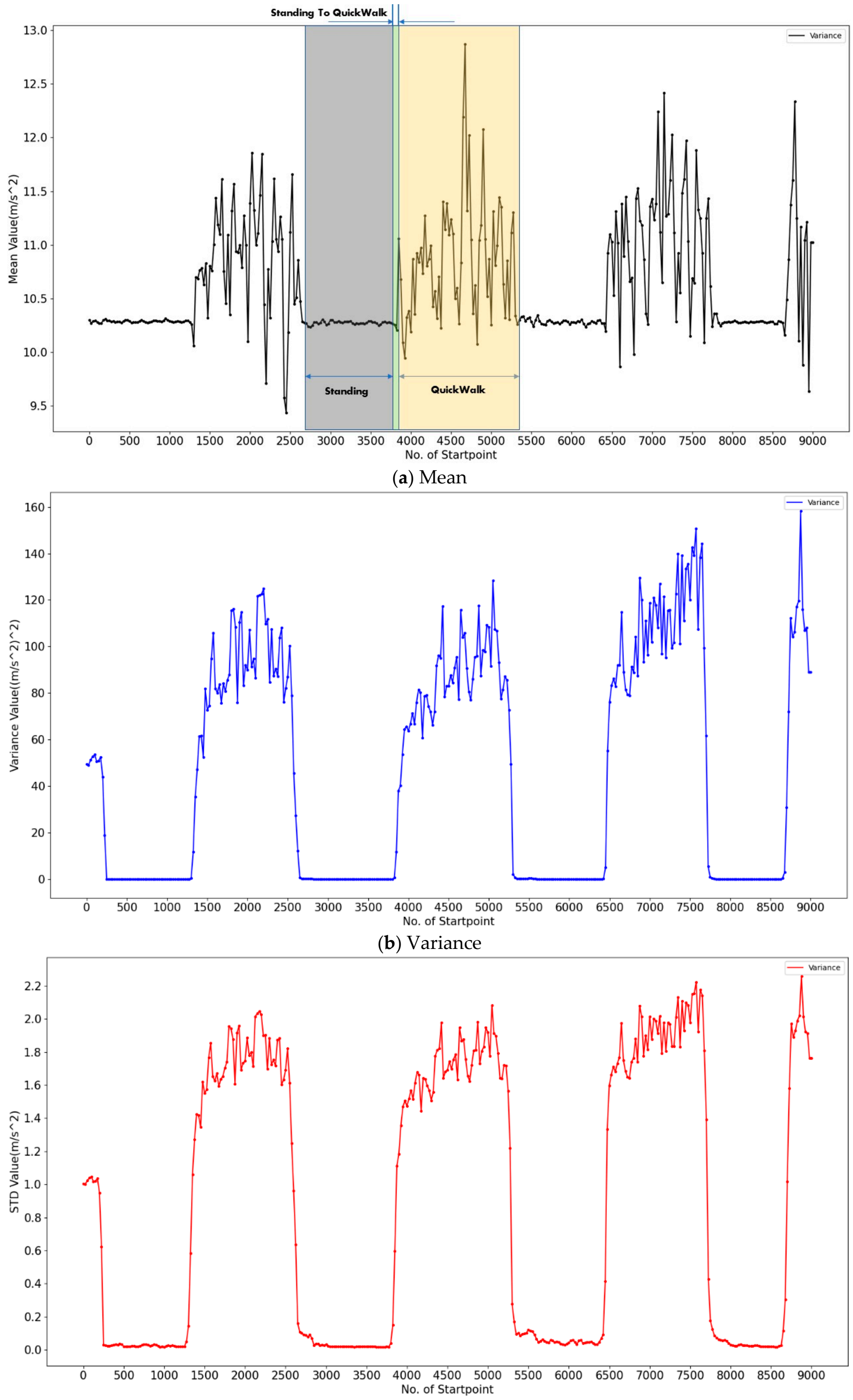
| No. | Feature | Formula |
|---|---|---|
| 1 | Mean | $\bar{a}$ |
| 2 | Variance | $\sum_{i=1}^{n}(a_i-\mu)^2$ |
| 3 | STD | $\sqrt{\frac{\sum_{i=1}^{n}(a_i-\mu)^2}{n}}$ |
| 4 | Maximum | $max(a_i)$ |
| 5 | Minimum | $min(a_i)$ |
| 6 | Range | $max(a_i)-min(a_i)$ |
| 7 | ZCR | $\sum_{i=1}^{n} sig(a_i>0)$ |
| 8 | Median | $median(a_i)$ |
| 9 | MAD | $median(\lvert a_i-median(a_i)\rvert)$ |
| 10 | Information Entropy | $-\sum_{i=1}^{m}(p_i*\log(p_i))$ |
| 11 | Kurtosis | $E\left[(a_i-\mu)^4/\sigma^4\right]$ |
| 12 | Skewness | $E\left[(a_i-\mu)^3/\sigma^3\right]$ |
| 13 | Coefficient | $cov(X,Y)$ |

ZCR: zero crossing rate MAD: absolute median difference.

We define $Acc_{all}$ as

$$Acc_{all} = \sqrt{Acc\_x^2 + Acc\_y^2 + Acc\_z^2}$$

In this stage, we focus on the $Acc_{all}$ trend of transition activity data. Figure 3 shows the trend of three main features of "Standing-QuickWalk".




(**a**) Mean

(**b**) Variance

(**c**) STD

**Figure 3.** Mean, variance and STD of "Standing-QuickWalk".

Take Figure 3a as an example. The data in the gray area are acquired when the user is standing. There are little changes over this period, and the small area in the middle represents the user stopping "Standing" and beginning to "QuickWalk", at which point the mean value increases significantly. Then, the data in the light-yellow area represent the user continuing to "QuickWalk", and the mean value changes dramatically in this part. It is obvious that the mean trends of these two larger regions are hugely different, so they can be easily distinguished. We observe the value range of these three kinds of features in Figure 3. Mean is in range [9,13], variance in the range [0, 160] and STD in the range [0,2.2]. Compared with the first two features, the STD is smaller and more controllable, which is convenient for us to fine-tune. Therefore, we decide to use the STD trend analysis (STD-TA) as the transition activity recognition method.

On the other hand, we choose SVM as the classifier for basic activities. As a common model in pattern recognition, SVM performs well in a number of studies. It does not rely on a tremendous amount of data input, which is suitable for research with a very small dataset. In our work, the feature vectors formed in the feature extraction stage are divided into two parts, one for training and the other for testing. After training SVM, we deploy it into a real-time environment. Every real data segment produces a probabilistic vector $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$, where $p_i$ represents the probability that the data segment belongs to the ith activity. We name $\mathbf{P}$ as the probabilistic result, which is a vital basis to judge whether the current activity is a transition one.

### 3.4. Transition Activity Recognition and Recognition Result

At this stage, we introduce the details of the STD-TA. First, we define the transition relationship of activities to determine where the transition activity can happen. We design a transition diagram, as Figure 4 shows.

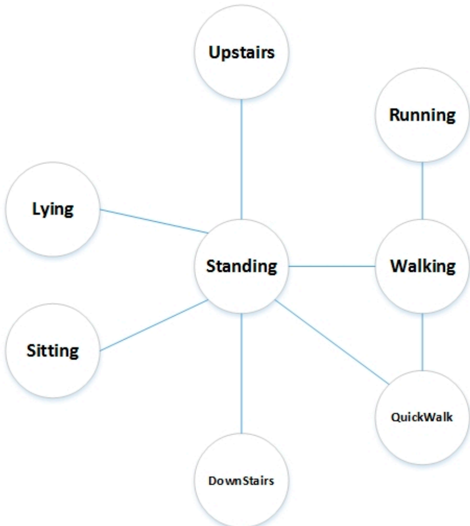



**Figure 4.** Activity transition diagram.

Figure 4 mainly specifies the basic activities between which the transition activities will take place, so that some illogical transitions could be effectively avoided to promote the overall accuracy. In the figure, there is no transition between basic activities without connection, such as "Lying" and "Running". The rules can effectively avoid some recognition errors. According to the diagram, there are 14 transition activities in total. Here, we clarify some definitions:

**Definition 1.**

$$\mathbf{FVList} = \{V_1, V_2, \ldots, V_n\}$$

*where* $V_i$ *is the feature vector extracted from the* $i$th *segment of a data sequence.*

**Definition 2.**

$$\mathbf{P} = \{\mathrm{SVM}(\mathrm{V}_1), \mathrm{SVM}(\mathrm{V}_2), \ldots, \mathrm{SVM}(\mathrm{V}_n)\},$$

$$\mathbf{PredAct} = \{\mathrm{index}(\max(\mathrm{P}_1)), \mathrm{index}(\max(\mathrm{P}_2)), \ldots, \mathrm{index}(\max(\mathrm{P}_n))\}$$

*where* $\mathrm{P}_i = \mathrm{SVM}(\mathrm{V}_i) \in \mathbf{P}$ *denotes the probabilistic results of* $\mathrm{V}_i$ *after being classified by* SVM, $\mathrm{PA}_i = \mathrm{index}(\max(\mathrm{P}_i)) \in \mathbf{PredAct}$ *is the category result of* $\mathrm{V}_i$, *which is the label with the maximum probability. We extract the* STD *value of* $\mathrm{V}_i$ *and record it as* $\mathrm{STD}_i$.

**Definition 3.**

$$\mathbf{Diff} = \{\mathrm{STD}_2\text{-}\mathrm{STD}_1, \mathrm{STD}_3\text{-}\mathrm{STD}_2, \ldots, \mathrm{STD}_n\text{-}\mathrm{STD}_{n-1}\}$$

Then, we depict the whole STD-TA algorithm as Algorithm 1 shows:

**Algorithm 1.** STD Trend Analysis Method (STD-TA).

Input: $\mathrm{PA}_{i-1}$, $\mathrm{Diff}_{i-1}$, $\mathrm{Diff}_i$, $\mathrm{Diff}_{i+1}$, $\mathrm{P}_i$, $\mathrm{STD}_i$, Intrans, Count

If $\mathrm{PA}_{i-1} \in$ *StaticActivity*:
If $\mathrm{STD}_i > 0.1$ and $\max(\mathrm{P}_i) < 0.9$:
Intrans = 1
Count = Count + 1
Else:
$\theta_1 = 1$ if $\mathrm{Diff}_i * \mathrm{Diff}_{i-1} > 0$ else 0
$\theta_2 = 1$ if $\mathrm{Diff}_{i+1} * \mathrm{Diff}_i > 0$ else 0
$\theta_3 = 1$ if $\mathrm{abs}(\mathrm{Diff}_i) > 0.1$ else 0
$\theta_4 = 0.6$ if $\max(\mathrm{P}_i) < 0.6$ else $\max(\mathrm{P}_i)$
$\mathrm{R} = 0.5 * \theta_1 + 0.4 * \theta_2 + 0.35 * \theta_3 - (1 - \theta_4) * 0.625$
If $\mathrm{R} \geq 1.22$:
Intrans = 1
Count = Count + 1
If Intrans == 1:
If $\mathrm{STD}_i \leq 0.1$ or $\max(\mathrm{P}_i) \geq 0.9$ or Count:
Intrans = 0
Count = 0
If Intrans == 1:
$\mathrm{PA}_i$ = TransitionActivity
Else:
$\mathrm{PA}_i = \mathrm{index}(\max(\mathrm{P}_i))$
Output: $\mathrm{PA}_i$

We refine all activities as Table 4 shows:

**Table 4.** Detailed category of activities.

| | | | |
|---|---|---|---|
| **Activity** | Basic Activity | Static Activity | Sitting Standing Lying |
| | | Dynamic Activity | Walking Upstairs Downstairs Running QuickWalk |
| | Transition Activity | Lying-Standing Sitting-Standing Standing-Upstairs Standing-Downstairs Standing-Walking Walking-Running Walking-QuickWalk | |

After refinement, activities are categorized as two kinds, basic activity and transition activity, and the basic activity mainly consists of static activity (SA) and dynamic activity (DA). Obviously, the SA are long-term stable postures, which means their STD values have few fluctuations.

In our algorithm, we first refer to the previous activity $PA_{i-1}$ of the current window. If it belongs to SA, then we observe $STD_i$. If it is a large value and the distribution of probabilistic results is scattered, then we consider the current window as transition activity. If $PA_{i-1} \in DA$, we take 4 factors into consideration: historical trend, future trend, real-time change and probabilistic results. $\theta_1$ denotes historical trend, and it covers the trend of the STD value of three windows. $\theta_1 > 0$ means the trend of $STD_{i\text{-}2}$, $STD_{i-1}$ and $STD_i$ are the same. $\theta_2$ denotes future trend, which needs to check the next STD value to judge the future trend. $\theta_2 > 0$ means the trend of $STD_{i-1}$, $STD_i$ and $STD_{i+1}$ are the same. $\theta_3$ is real-time change. When its value increases sharply and overcomes a threshold, it suggests that the current window suffers a huge change with respect to the previous window. $\theta_4$ denotes probabilistic results. When its value becomes too small, it means that SVM cannot clearly recognize the activity of the current window, which suggests that the current window is likely to be a transition activity. These four factors are a vital basis for judging if the current window is a transition activity. Here we define $\theta = \{ \theta_1, \theta_2, \theta_3, \theta_4\}$, and we give the detail determination process of its value in Section 4.

The algorithm finally outputs the recognition result. The transition activities are given the same label, while if it is judged as a basic activity, it will be further classified by SVM to determine its final activity category.

## 4. Experiments

### 4.1. Android Application

We develop a simple Android application (APP) and deploy it in an Mi phone to conduct our experiments. The APP implements the whole architecture and we deploy it on the phone to carry out data collection and real-time activity recognition. Figure 5 shows the user interface (UI) of APP.

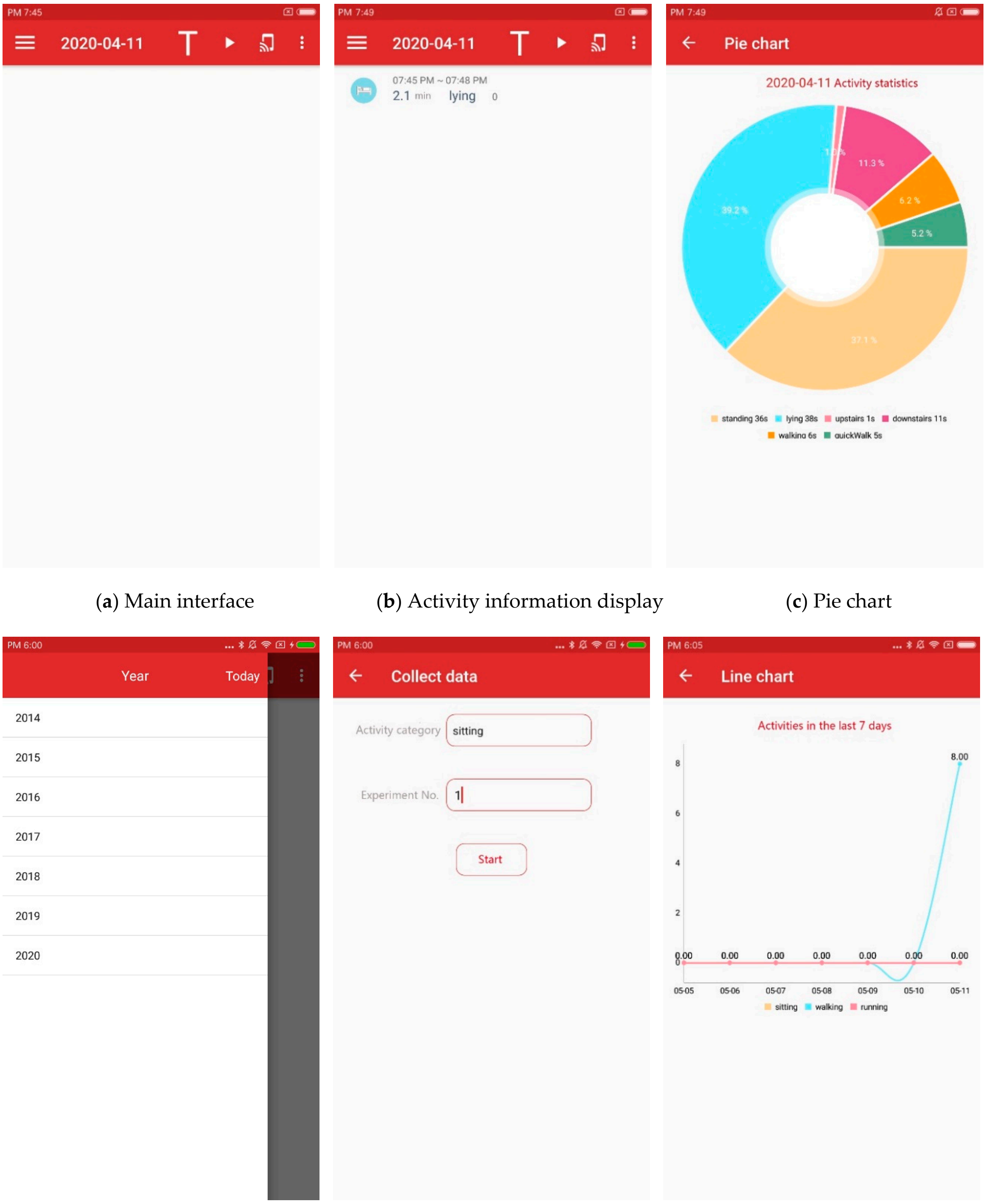
We design this APP to realize the activity recognition and statistics of subjects. We collect activity data using the built-in accelerometer and barometer of a smartphone and recognize the daily activity via the model described in the former sections. Due to the fact that the APP is just in a test stage, the recognition results are stored locally. Meanwhile, we design a long-term monitoring module for users in the APP. The line chart in Figure 5 gives the users' activities in the previous 7 days, while the pie chart shows the duration and proportion of each daily activity of the users. These functions provide vital context information for high-level applications.

The entire APP is only a test version at present and still needs to be further improved. In the future, we will consider transferring the recognition model and results to the cloud to save the resources of the smart device.

### 4.2. Experiment 1: Classifier Comparison

After data collection and segmentation, we divide the whole dataset into 10 parts, and 10-fold cross validation is carried out on the classifiers. We select three common models, decision tree (DT) [31], K-nearest neighbor (KNN) [32] and support vector machine (SVM) [33], to carry out this experiment and compare the final results. We only consider the characteristics of these classifiers, so here the target activities only include basic activities. Table 5 shows the test results.

According to the results, all classifiers have tremendous high recognition accuracy for static activities (A01–A03), which is nearly 100%. However, this does not mean that it can maintain a high level under real scenarios. For A04, A07 and A08, the accuracy of the three models is all over 95%, while SVM is still slightly better than the other two. Although we combine the data of the accelerometer and barometer, these classifiers are not so effective on A05 and A06, and their accuracy is around 75%. Meanwhile, SVM performs better than the other two on DA. Lastly, we choose SVM as the core classifier in our work.




(**a**) Main interface (**b**) Activity information display (**c**) Pie chart

(**d**) Date selection (**e**) Data collecting (**f**) Line chart

**Figure 5.** UI of the Android application.

**Table 5.** Recognition results comparison between three classifiers.

| DT | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 |
|---|---|---|---|---|---|---|---|---|
| **A01** | 703 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A02** | 0 | 745 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A03** | 0 | 0 | 711 | 0 | 0 | 0 | 0 | 0 |
| **A04** | 0 | 0 | 0 | 664 | 4 | 11 | 0 | 14 |
| **A05** | 0 | 0 | 0 | 18 | 478 | 64 | 0 | 4 |
| **A06** | 0 | 6 | 0 | 33 | 63 | 339 | 0 | 3 |
| **A07** | 0 | 0 | 0 | 1 | 0 | 0 | 688 | 0 |
| **A08** | 0 | 0 | 0 | 4 | 1 | 1 | 0 | 692 |
| **KNN** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** |
| **A01** | 703 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A02** | 0 | 745 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A03** | 0 | 0 | 711 | 0 | 0 | 0 | 0 | 0 |
| **A04** | 0 | 0 | 0 | 679 | 4 | 4 | 0 | 6 |
| **A05** | 0 | 0 | 0 | 48 | 484 | 22 | 0 | 10 |
| **A06** | 0 | 0 | 0 | 51 | 39 | 351 | 0 | 3 |
| **A07** | 0 | 0 | 0 | 0 | 0 | 0 | 689 | 0 |
| **A08** | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 693 |
| **SVM** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** |
| **A01** | 703 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A02** | 0 | 745 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A03** | 0 | 0 | 711 | 0 | 0 | 0 | 0 | 0 |
| **A04** | 0 | 0 | 0 | 684 | 1 | 3 | 0 | 5 |
| **A05** | 0 | 0 | 0 | 12 | 516 | 30 | 0 | 6 |
| **A06** | 0 | 6 | 0 | 15 | 39 | 381 | 0 | 3 |
| **A07** | 0 | 0 | 0 | 0 | 0 | 0 | 689 | 0 |
| **A08** | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 695 |

A01–A08 denote basic activities “sitting, standing, lying, walking, upstairs, downstairs, running, quickwalk”.

### 4.3. *Experiment 2: Determination of Vector θ*

According to Algorithm 1, we recognize the transition activity by analyzing the trend of STD. For SA, its STD value is stable and fluctuates around 0, which is easy to distinguish from DA. However, the boundary between DA and transition activity is confusing, which is the reason why we define the factor vector $\theta$.

We take a transition activity as an example. Figure 6 shows the data fragment of “Standing-Walking”.

According to the figure, the vertical axis represents the STD value of the window, and the horizontal axis denotes the starting data point of the window. For example, there is a data point ($x = 1150$, $y = 0.0159$) in the figure, which represents the data of a window composed of 50 data points 1150–1200. The STD value of these data is 0.0159.

In the first half of this clip, this subject stays standing, and the STD values of these windows fluctuate around 0 with few changes. When $x = 1350$, the subject starts to walk. After $x = 1450$, the subject is completely in walking state. We can clearly detect the change of body posture from the trend of the STD value, especially when $x \in [1350, 1425]$. This is the transition we need to recognize. There are several distinct features of these STD values:

1. Continuous and identical changes. For example, the STD shows a monotonic increasing trend when $x \in [1350, 1425]$. Therefore, for the ith window ($i = x/25$), we check $y_{i-1} - y_{i-2}$, $y_i - y_{i-1}$ and $y_{i+1} - y_i$ to judge the current STD trend. If the three values have the same symbol, which means they are all positive or negative, it is likely that the current activity is a transition one. This is what $\theta_1$ and $\theta_2$ represent.

2. Huge change. Transition activity is a transition from one stable state to another, so the change between them is usually intense. Thus, for the ith window (i = x/25), we check | $y_i$-$y_{i-1}$ |, when its value exceeds the threshold (here we set threshold as 0.1), we judge that the current window may be in transition. This is what $\theta_3$ represents. It should be noted that the amount of DA data also meets this condition. However, it is still an important indicator of transition activity.
3. Uncertain result. Due to the fact that the transition activity has the same features as multi activities, the recognition result is usually uncertain, although this situation is not absolute. Thus, for the ith window (i = x/25), we check max(P), which is the maximum probability that the current activity belongs to some basic activity. This is what $\theta_4$ represents.

We extract all the transition activity data fragments and the same amount of basic activity data fragments. For each segment, we calculate the $\theta$ as Algorithm 1 depicts. Meanwhile, we set labels for each fragment, 0 for basic activity and 1 for transition activity. Multiple linear regression (MLR) is utilized. The problem is described as follows:


**Figure 6.** A fragment of "Standing-Walking".

This act denotes current activity, ∀ **W** = {w1,w2,w3,w4},

$$\mathbf{W}\theta^{T} = \begin{cases} 0 & \textit{ThisAct} \in \textit{Basic Activity} \\ 1 & \textit{ThisAct} \in \textit{Transition Activity} \end{cases}$$

The function of MLR is finding a proper **W**. With this approach, we get:

$$\mathbf{W} = \{0.5139511, 0.4159489, 0.35475093, -0.61839164\}$$

After a fine-tuning, we finally determine **W** as Algorithm 1 shows.

*4.4. Experiment 3: Overall Performance*

In this part, we explore the overall performance of our proposed method. For comparison, we set up control groups. Referring to the idea in [34], we regard all transition activities as one activity and assign a unified label to them. We then use SVM and KNN as classifiers for both transition and basic activities. Finally, we do the comparison with our method after experiments.

Here, we first use the whole dataset we acquired before, including basic and transition activities, to train the KNN and SVM. Moreover, we introduce the CUSUM chart (cumulative sum control chart) as another state-of-the-art algorithm in this experiment. CUSUM has excellent performance in change point detection issues [35,36], which are similar with our problem. We use a similar strategy using SVM as the basic classifier and CUSUM as the transition activity recognition method. Then, we invite five new subjects to perform activities as a test set for the four methods. Table 6 gives the results.

**Table 6.** Overall performance of four methods.

| **SVM** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** | **A09** | **SUM** | **Recall** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **A01** | 6552 | 34 | 0 | 44 | 0 | 0 | 0 | 0 | 0 | 6630 | 0.988235 |
| **A02** | 0 | 6417 | 25 | 33 | 0 | 0 | 0 | 0 | 0 | 6475 | 0.991042 |
| **A03** | 5 | 5 | 6510 | 0 | 0 | 0 | 0 | 0 | 0 | 6520 | 0.998466 |
| **A04** | 0 | 2 | 114 | 5621 | 110 | 123 | 84 | 125 | 246 | 6425 | 0.874864 |
| **A05** | 0 | 0 | 0 | 243 | 5966 | 117 | 43 | 19 | 27 | 6415 | 0.930008 |
| **A06** | 0 | 0 | 0 | 108 | 123 | 5852 | 120 | 86 | 196 | 6485 | 0.90239 |
| **A07** | 0 | 0 | 0 | 32 | 0 | 13 | 6354 | 56 | 0 | 6455 | 0.984353 |
| **A08** | 0 | 0 | 0 | 54 | 51 | 25 | 71 | 6146 | 66 | 6413 | 0.958366 |
| **A09** | 0 | 0 | 0 | 352 | 186 | 174 | 76 | 213 | 1944 | 2945 | 0.660102 |
| **KNN** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** | **A09** | **SUM** | **Recall** |
| **A01** | 6617 | 0 | 8 | 5 | 0 | 0 | 0 | 0 | 0 | 6630 | 0.998039 |
| **A02** | 0 | 6475 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6475 | 1 |
| **A03** | 4 | 12 | 6504 | 0 | 0 | 0 | 0 | 0 | 0 | 6520 | 0.997546 |
| **A04** | 0 | 67 | 19 | 5552 | 76 | 114 | 24 | 169 | 404 | 6425 | 0.864125 |
| **A05** | 0 | 0 | 22 | 188 | 5644 | 139 | 213 | 114 | 95 | 6415 | 0.879813 |
| **A06** | 0 | 0 | 13 | 355 | 345 | 5124 | 213 | 267 | 168 | 6485 | 0.790131 |
| **A07** | 0 | 0 | 0 | 51 | 67 | 0 | 6289 | 41 | 7 | 6455 | 0.974284 |
| **A08** | 0 | 0 | 0 | 19 | 0 | 0 | 12 | 6312 | 70 | 6413 | 0.984251 |
| **A09** | 0 | 0 | 0 | 220 | 206 | 162 | 167 | 245 | 1945 | 2945 | 0.660441 |
| **STD-TA** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** | **A09** | **SUM** | **Recall** |
| **A01** | 6618 | 4 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 6630 | 0.99819 |
| **A02** | 0 | 6475 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6475 | 1 |
| **A03** | 22 | 41 | 6457 | 0 | 0 | 0 | 0 | 0 | 0 | 6520 | 0.990337 |
| **A04** | 0 | 0 | 22 | 6014 | 67 | 0 | 122 | 67 | 133 | 6425 | 0.936031 |
| **A05** | 0 | 0 | 0 | 154 | 5848 | 46 | 77 | 41 | 249 | 6415 | 0.911613 |
| **A06** | 0 | 0 | 0 | 188 | 205 | 5622 | 169 | 105 | 196 | 6485 | 0.866924 |
| **A07** | 0 | 0 | 0 | 10 | 4 | 0 | 6375 | 22 | 44 | 6455 | 0.987607 |
| **A08** | 0 | 0 | 0 | 70 | 14 | 0 | 42 | 6254 | 33 | 6413 | 0.975207 |
| **A09** | 0 | 0 | 7 | 122 | 105 | 91 | 78 | 25 | 2517 | 2945 | 0.854669 |
| **CUSUM** | **A01** | **A02** | **A03** | **A04** | **A05** | **A06** | **A07** | **A08** | **A09** | **SUM** | **Recall** |
| **A01** | 6603 | 17 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 6630 | 0.995928 |
| **A02** | 0 | 6475 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6475 | 1 |
| **A03** | 6 | 3 | 6511 | 0 | 0 | 0 | 0 | 0 | 0 | 6520 | 0.99862 |
| **A04** | 0 | 21 | 1 | 5741 | 72 | 72 | 88 | 76 | 354 | 6425 | 0.893541 |
| **A05** | 0 | 0 | 12 | 121 | 5798 | 139 | 109 | 114 | 122 | 6415 | 0.903819 |
| **A06** | 0 | 0 | 13 | 241 | 345 | 5336 | 154 | 267 | 129 | 6485 | 0.822822 |
| **A07** | 0 | 0 | 0 | 25 | 44 | 57 | 6300 | 24 | 5 | 6455 | 0.975988 |
| **A08** | 0 | 0 | 2 | 31 | 20 | 40 | 51 | 6147 | 122 | 6413 | 0.958522 |
| **A09** | 0 | 0 | 0 | 120 | 206 | 162 | 167 | 133 | 2157 | 2945 | 0.732428 |

According to the results in the table, the three models all perform well on basic activities (A01–A08). Due to the fact that the SAs are stable and easy to distinguish from each other, there are few errors that happen when recognizing them (A01–A03). Accuracy on "Walking" (A04) is relatively low, because as a basic DA, its behavior mode is easily confused with other kinds of DA. In the recognition of "Upstairs" (A05) and "Downstairs" (A06), SVM, STD-TA and CUSUM, which also use SVM as core classifier, have a 6%–10% higher accuracy than KNN. This is similar to our results in Experiment 1. For "Running" (A07) and "QuickWalk" (A08), the results are all at a high level, above 95%. We believe that the intensity of these two activities is tremendously high, which leads to them being quite different from other activities and easy to distinguish.

A09 represents all transition activities. Both SVM and KNN maintain an accuracy of 66%. Obviously, a number of pieces of data which belong to the transition activity category are mistakenly identified as basic activities, and vice versa. We infer that transition activity has high similarity with

DA. The CUSUM achieves a recognition accuracy of 73.24%. It is effective for detecting the change point of the activity sequence. However, this method is effective for a single kind of transition activity, which has limitation on recognizing multiple kinds of transitions in real scenarios. In contrast, our method improves the accuracy of detection of transition activity by nearly 20%, reaching 80%. It is supposed to be the trend features extracted by the STD-TA that produces the promotion.

## 5. Conclusions and Future Work

HAR is faced with some problems, and the transition activity mentioned in this paper is one of them. To solve this issue, we designed an algorithm based on STD trend analysis. For basic activity, we mainly utilized SVM for recognition. For transition activity, we analyzed the STD value of data to judge the trend of the overall data flow to recognize the activity. Through result comparison, the accuracy of our algorithm is 20% higher, reaching 82.85%, than only using a machine learning model in identifying the transition activity.

Our work has achieved promising results in distinguishing transition activity from basic activity. However, our method does not have an absolute advantage over the other two classifiers in the control group on distinguishing between transition activity and walking. In addition, the generalization of some specific parameters we calculate in this paper, such as $\mathbf{W}$ and $\theta$, is limited. If there were some users whose height or age is beyond the range of the training set, the accuracy would be affected. Therefore, we should design an incremental update mechanism for these parameters in our future work. Moreover, we do not further discuss the specific categories of transition activities. This is because that we can judge the specific type of transition activity through activities before and after it. Adding the specific recognition method could increase the complexity of the whole architecture. In future work, we will focus on the generalization of our model and discuss more complex daily activities (e.g., cooking, reading, etc.).

**Author Contributions:** Conceptualization, J.S. and D.Z.; methodology, J.S.; software, J.S.; validation, Z.Z. and D.Z.; formal analysis, Z.Z.; investigation, J.S.; resources, Z.Z. and D.Z.; data curation, J.S.; writing—original draft preparation, J.S.; writing—review and editing, J.S. and D.Z.; visualization, J.S.; supervision, Z.Z.; project administration, D.Z.; funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National High-tech R&D Program of China, grant number 2013AA01A215 and the APC was funded by the National High-tech R&D Program of China, grant number 2013AA01A215.

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

## References

1. Cornacchia, M.; Zheng, Y.; Velipasalar, S.; Ozcan, K. A Survey on Activity Detection and Classification Using Wearable Sensors. *IEEE Sens. J.* **2016**, *17*, 386–403. [CrossRef]
2. Chen, L.-W.; Ho, Y.-F.; Kuo, W.-T.; Tsai, M.-F. Intelligent file transfer for smart handheld devices based on mobile cloud computing. *Int. J. Commun. Syst.* **2015**, *30*, e2947. [CrossRef]
3. Altun, K.; Barshan, B. Human Activity Recognition Using Inertial/Magnetic Sensor Units. *Appl. Evol. Comput.* **2010**, *6219*, 38–51. [CrossRef]
4. Hyun, W.; You, I.; Jang, J.; Leu, F.-Y. A Wireless Body Sensor Network and Its Applications: Rehearsal with a Smartphone. In Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Fukuoka, Japan, 6–8 July 2016; pp. 415–418.
5. Chetty, G.; White, M. Body sensor networks for human activity recognition. In Proceedings of the 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 11–12 February 2016; pp. 660–665.
6. Espinilla, M.; Medina, J.; Hallberg, J.; Nugent, C. A new approach based on temporal sub-windows for online sensor-based activity recognition. *J. Ambient Intell. Humaniz. Comput.* **2018**, 1–13. [CrossRef]
7. Garcia-Ceja, E.; Galván-Tejada, C.E.; Brena, R. Multi-view stacking for activity recognition with sound and accelerometer data. *Inf. Fusion* **2018**, *40*, 45–56. [CrossRef]

8. Zahin, A.; Tan, L.T.; Hu, R.Q. Sensor-Based Human Activity Recognition for Smart Healthcare: A Semi-supervised Machine Learning. In *Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*; Springer Science and Business Media LLC: Harbin, China, 2019; pp. 450–472.
9. Yao, R.; Lin, G.; Shi, Q.; Ranasinghe, D. Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. *Pattern Recognit.* **2018**, *78*, 252–266. [CrossRef]
10. Shen, C.; Chen, Y.; Yang, G. On motion-sensor behavior analysis for human-activity recognition via smartphones. In Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), Sendai, Japan, 29 February–2 March 2016; pp. 1–6.
11. Reyes-Ortiz, J.-L.; Oneto, L.; Samà, A.; Parra, X.; Anguita, D. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **2016**, *171*, 754–767. [CrossRef]
12. Noor, M.H.M.; Salcic, Z.; Wang, K.I.-K. Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive Mob. Comput.* **2017**, *38*, 41–59. [CrossRef]
13. Pansiot, J.; Stoyanov, D.; McIlwraith, D.; Lo, B.; Yang, G.Z. Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems. In *World Congress on Medical Physics and Biomedical Engineering 2006*; Springer Science and Business Media LLC: Seoul, Korea, 2006; Volume 13, pp. 208–212.
14. Minnen, D.; Westeyn, T.; Ashbrook, D.; Presti, P.; Starner, T. Recognizing Soldier Activities in the Field. In *World Congress on Medical Physics and Biomedical Engineering 2006*; Springer Science and Business Media LLC: Seoul, Korea, 2007; Volume 13, pp. 236–241.
15. Akhavian, R.; Behzadan, A.H. Smartphone-based construction workers' activity recognition and classification. *Autom. Constr.* **2016**, *71*, 198–209. [CrossRef]
16. Wang, R.; Chen, F.; Chen, Z.; Li, T.; Harari, G.; Tignor, S.; Zhou, X.; Ben-Zeev, D.; Campbell, A.T. StudentLife. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan, 7–11 September 2014; pp. 3–14. [CrossRef]
17. Bisio, I.; Lavagetto, F.; Marchese, M.; Sciarrone, A. Smartphone-based user activity recognition method for health remote monitoring applications. In Proceedings of the International Conference on Pervasive and Embedded Computing and Communication Systems, Rome, Italy, 24–26 February 2012; pp. 200–205.
18. Ronao, C.A.; Cho, S.-B. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In Proceedings of the 2014 10th International Conference on Natural Computation (ICNC), Xiamen, China, 19–21 August 2014; pp. 681–686.
19. Lu, D.-N.; Nguyen, T.-T.; Ngo, T.-T.-T.; Nguyen, T.-H.; Nguyen, H.-N.; Akagi, M.; Nguyen, T.-T.; Vu, D.-T.; Phung, T.-N.; Huynh, V.-N. Mobile Online Activity Recognition System Based on Smartphone Sensors. *Adv. Intell. Syst. Comput.* **2016**, *538*, 357–366. [CrossRef]
20. Wu, H.; Pan, W.; Xiong, X.; Xu, S. Human activity recognition based on the combined SVM&HMM. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 219–224.
21. Fan, L.; Wang, Z.; Wang, H. Human Activity Recognition Model Based on Decision Tree. In Proceedings of the 2013 International Conference on Advanced Cloud and Big Data, Nanjing, China, 13–15 December 2013; pp. 64–68.
22. Feng, Z.; Mo, L.; Li, M. A Random Forest-based ensemble method for activity recognition. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; Volume 2015, pp. 5074–5077.
23. Song, M.H.; Lee, Y.H. Direct optimization of inference model for human activity and posture class recognition. *Multimed. Tools Appl.* **2015**, *74*, 1–18. [CrossRef]
24. Aminikhanghahi, S.; Cook, D.J. Using change point detection to automate daily activity segmentation. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017.
25. Ali, R.; Atallah, L.; Lo, B.; Yang, G.-Z. Transitional Activity Recognition with Manifold Embedding. In Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks, Berkeley, CA, USA, 3–5 June 2009; pp. 98–102. [CrossRef]
26. Huynh-The, T.; Hua, C.-H.; Tu, N.A.; Hur, T.; Bang, J.; Kim, D.; Amin, M.B.; Kang, B.H.; Seung, H.; Shin, S.-Y.; et al. Hierarchical topic modeling with pose-transition feature for action recognition using 3D skeleton data. *Inf. Sci.* **2018**, *444*, 20–35. [CrossRef]

27. Shi, J.; Zuo, D.; Zhang, Z.; Luo, D. Sensor-based activity recognition independent of device placement and orientation. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3823. [CrossRef]
28. Madevska-Bogdanova, A.; Nikolik, D.; Curfs, L. Probabilistic SVM outputs for pattern recognition using analytical geometry. *Neurocomputing* **2004**, *62*, 293–303. [CrossRef]
29. Liu, Y.; Wen, K.; Gao, Q.; Gao, X.; Nie, F. SVM based multi-label learning with missing labels for image annotation. *Pattern Recognit.* **2018**, *78*, 307–317. [CrossRef]
30. Zhu, C.; Sheng, W. Human daily activity recognition in robot-assisted living using multi-sensor fusion. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 303–304.
31. Kumar, R.; Singh, B.; Shahani, D.T.; Chandra, A.; Al-Haddad, K.; Garg, R. Recognition of Power-Quality Disturbances Using S-Transform-Based ANN Classifier and Rule-Based Decision Tree. *IEEE Trans. Ind. Appl.* **2014**, *51*, 1249–1258. [CrossRef]
32. Chen, S.-B.; Xu, Y.-L.; Ding, C.H.; Luo, B. A Nonnegative Locally Linear KNN model for image recognition. *Pattern Recognit.* **2018**, *83*, 78–90. [CrossRef]
33. Codella, N.; Cai, J.; Abedini, M.; Garnavi, R.; Halpern, A.; Smith, J.R. Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images. In *Applications of Evolutionary Computation*; Springer Science and Business Media LLC: Munich, Germany, 2015; Volume 9352, pp. 118–126.
34. Kozina, S.; Gjoreski, H.; Gams, M.; Luštrek, M. Three-layer Activity Recognition Combining Domain Knowledge and Meta-classification. *J. Med. Biol. Eng.* **2013**, *33*, 406–414. [CrossRef]
35. Du Nguyen, H.; Tran, K.P.; Heuchenne, H.L. CUSUM control charts with variable sampling interval for monitoring the ratio of two normal variables. *Qual. Reliab. Eng. Int.* **2019**, *36*, 474–497. [CrossRef]
36. Aytaçoğlu, B.; Woodall, W.H. Dynamic probability control limits for CUSUM charts for monitoring proportions with time-varying sample sizes. *Qual. Reliab. Eng. Int.* **2019**, *36*, 592–603. [CrossRef]