IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 1, 2017, accepted January 19, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2676168

Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition

YUFEI CHEN AND CHAO SHEN, (Member, IEEE)
MOE Key Laboratory for Intelligent Networks and Network Security, State Key Laboratory for Manufacturing Systems, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: C. Shen (cshen@sei.xjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61403301 and Grant 61221063, in part by the China Postdoctoral Science Foundation under Grant 2014M560783 and Grant 2015T81032, in part by the Natural Science Foundation of Shaanxi Province under Grant 2015JQ6216, in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), and in part by the Fundamental Research Funds for the Central Universities under Grant xjj2015115.

ABSTRACT The proliferation of smartphones has significantly facilitated people's daily life, and diverse and powerful embedded sensors make smartphone a ubiquitous platform to acquire and analyze data, which may also provide great potential for efficient human activity recognition. This paper presents a systematic performance analysis of motion-sensor behavior for human activity recognition via smartphones. Sensory data sequences are collected via smartphones, when participants perform typical and daily human activities. A cycle detection algorithm is applied to segment the data sequence for obtaining the activity unit, which is then characterized by time-, frequency-, and wavelet-domain features. Then both personalized and generalized model using diverse classification algorithms are developed and implemented to perform activity recognition. Analyses are conducted using 27 681 sensory samples from 10 subjects, and the performance is measured in the form of F-score under various placement settings, and in terms of sensitivity to user space, stability to combination of motion sensors, and impact of data imbalance. Extensive results show that each individual has its own specific and discriminative movement patterns, and the F-score for personalized model and generalized model can reach 95.95% and 96.26%, respectively, which indicates our approach is accurate and efficient for practical implementation.

INDEX TERMS Smartphone, motion sensor, behavior analysis, human activity recognition, performance analysis.

I. INTRODUCTION

Recent years have witnessed an increasing development of smartphones, which become more and more ubiquitous and popular. Thanks to the tremendous growth of computing and sensing power, many applications like health monitor, game, sports tracking, are able to be implemented on smartphones. As an omnipresent computing and data collection platform, smartphone has inspired many researches on indoor pedestrian tracking [1], [2], human activity recognition [3], [4], authentication based on biological characteristics [5], [6], and etc. In this paper, we mainly focus on the human-activity recognition approach based on motion sensor via smartphones.

Among recent studies focusing on sensor-based human activity recognition, the analysis of accelerometer data attracts the most attention, in which most researches chose waist as the position to carry smartphones [7], [8]. However, unlike wearable devices (e.g., smartwatch and smartband), smartphones are carried in various and uncertain positions in our daily life. We may hold the phone in our hand, and then put it into our jacket or trousers pocket. Even if when we are jogging or doing other sports, the phone is always carried at our upper arm. This may be one of main reasons why it is hard to conduct human activity recognition using smartphone sensors. So far, only a few researches consider the impact of the smartphone-placement [9], [25], [26], [29].

This paper presents a framework and performance analysis of smartphone-sensor based human activity recognition, which can continuously perceive user's motion state. Sensory data (mainly including accelerometer and gyroscope sensors) were collected when participants perform some typical and daily human activities: descending stairs, ascending stairs, walking, jogging and jumping. We then extracted every single movement unit from the long-lasting data sequence by using cycle detection algorithm, and then we obtained time-, frequency-, and wavelet-domain features from the segmented data. Moreover, an empirical study was conducted to measure the discriminability of these features according to the Kolmogorov-Smirnov test. In this paper, we applied three classification techniques to conduct activity recognition task: Nearest Neighbors, Random Forests and Support Vector Machines. To evaluate the stability and flexibility of our proposed approach, we considered five positions to carry smartphone, each of which is commonly used in our daily life or mostly adopted in sensor-based human activity recognition research. What's more, we built up three training models from 10 recruited subjects with 27,681 movement units: one personalized model and two generalized models, depending on whether the training/testing data came from a single user or from multiple users. We also investigated how the combination of sensors would affect the recognition performance. Considering the fact that in the actual scenario of human daily life, different activities have different probabilities of occurrence (e.g., compared with walking, jumping occurs less frequently in one's daily life), here we study the impact of the data balance. Our main purpose of this paper is to make a systematic analysis of the sensor-behavior based human- activity recognition via smartphones, and our approach is not a sole but a complementary technique to improve the accuracy in sensor-based activity recognition. The main contributions of this paper can be summarized as four folds:

First, we lay an empirical study of analyzing smartphone motion-sensor data for human activity recognition. From the data collection, data preprocessing, feature analysis, classifier implementation to activity recognition, we provide an exhaustive approach roadmap for sensor-based human activity recognition, with the hope to give some inspirations for the related research field.

Second, different activities in real scenarios would have different probabilities of occurrence in human daily life. We take the impact of imbalanced dataset into consideration, and employ the oversampling technique to reduce such effects, to enhance our approach's flexibility in practical applications.

Third, we propose a set of sensor-behavior features by characterizing various operational properties, including time-, frequency- and wavelet-domain features. We also conduct an empirical analysis of the sensor-behavior features by using two-sample K-S test. We then measure the stability and discriminability of these features, to give us an in-depth understanding how the sensor-behavior features reflect different activities' characteristics.

Fourth, to the best of our knowledge, few studies have examined how user space affects the performance in the training stage, which has to be taken into account for practical applications. Here we built up three classification models (one personalized model and two generalized models) to evaluate the recognition performance.

In general, this study systematically analyzes various factors that may influence the recognition performance, including the positions carrying the smartphone, the user space, the degree of data imbalance and the combination of motion sensors, to provide a good supplement to existing activity recognition mechanism.

The remainder of this paper is organized as follows: Section II introduces the background, and reviews related work. Section III offers details of the data acquisition and preprocessing procedure. Section IV describes the sensor-behavior analysis and feature extraction method. Section V briefly introduces the classifier implementation. Section VI depicts the evaluation methodology. Section VII offers the evaluation results in terms of recognition accuracy, flexibility to various placement settings, sensitivity to motion sensors, and impact of data balance. Section VIII provides a discussion, and Section IX concludes.

II. BACKGROUND AND RELATED WORK

In this section, we briefly introduce researches on human activity recognition. Then we focus on applying smartphone sensor data to activity recognition.

A. BACKGROUND OF HUMAN ACTIVITY RECOGNITION

The aim of human activity recognition is to discover human physical activity patterns by analyzing the movement data captured via multiple sensors. Human activity recognition has several potential applications, which can not only monitor human activity (e.g., fall-down detection, health-care), but also help to better understand human activity (e.g., surveillance environments, which can automatically detect abnormal activities). Currently, the activity recognition methods can be mainly summarized as two categories: vision-based and sensor-based.

Vision-based technique appeared earlier than the sensor-based technique. Its core processing stages mainly include data preprocessing, object segmentation, feature extraction and classifier implementation [11]–[13]. Though many efficient techniques have been proposed in the past few decades, vision-based HAR still remains challenging: the position and angle of the observer, the object's body size and clothes, the color of backgrounds and the light intensity will all affect the accuracy.

With the rapid development in the MEMS (Microelectromechanical Systems), inertial sensors become smaller, lighter, less expensive and more accurate. Compared with video-based method, sensor-based method is more robust in various environments, and the devices are cheaper and lighter. Moreover, many smartphone and smart wearable devices are equipped with multiple sensors (e.g., accelerometer and gyroscope), making sensors easily-acquirable. With these advantages, sensor-based activity recognition attracts more attentions in recent years. Previous studies usually used more than one sensor to recognize human activities: Farringdom *et al.* [14] used a sensor jacket equipped with several accelerometers to discriminate three types of

TABLE 1. Common sensors in popular smartphones.

Sensor	Sony Xperia Z5	iPhone6	Samsung Galaxy S6
Accelerometer	√	√	√
Gyroscope	√	√	√
Light	√	√	√
Proximity	√	√	√
Barometer	√	√	√
GPS	√	√	√

static activities (i.e., sitting, standing and lying) and two type of dynamic activities (i.e., walking and running). Mantyjarvi *et al.* [15] attached two accelerometers at subject's left and right sides of the hip to recognize level walking, walking up, walking down and opening doors. To make it more feasible for long-term monitoring, some researches have investigated the approach only using a single sensor [16]–[18]. Besides, features and classification algorithms are also various. Generally, features can be divided into two groups: time-domain and frequency-domain [19], while the mostly used classifiers are Support Vector Machines [20], Neural Network [21], K-Nearest-Neighbor [22] and Hidden Markov Model [23].

B. HUMAN ACTIVITY RECOGNITION VIA SMARTPHONE SENSORS

Nowadays, sensing and computing capabilities have become a standard on current smartphones, and researchers have begun to use smartphone as the experimental platform to conduct sensor-based activity recognition. Table 1 lists some common sensors equipped in popular smartphones. In this paper, we mainly focus on two motion sensors: accelerometer and gyroscope.

Compared with recognition mechanisms using wearable sensors, smartphone-based activity recognition has faced more problems and challenges. One of the biggest problems is the variety of smartphone's position and orientation. The phone may be held in hand, and also may be put into pocket. Another problem is that human's activity is always a complex procedure, including the rotation of joints and the contraction of muscles, however, we can only get only one part of body's motion data at a time if without other auxiliary sensors, making smartphone-sensor-based activity recognition a difficult task. Yet, due to the limit of cost, sensors' accuracy tends to be not ideal. The collected data are often mixed with noise. To make their approaches more feasible, researches mainly focused on de-nosing, feature construction, selection of carrying position and sensors, and detector implementation. Table 2 presents a brief summary of previous work.

Brezmes *et al.* [24] provided a primary result using accelerometer data on a mobile phone for activity recognition. In the training stage, the user should train the application in his usual way to carry the phone, but the location was not restricted, no matter a chest pocket, a front trousers pocket, and etc. This work covered 6 daily activities, and a k-NN classifier was implemented. Though its recognition algorithm was simple, it was quite accurate, which claimed the highest accuracy achieved 90%, and the lowest achieved 70%.

Henpraserttae *et al.* [25] presented some results for different orientations and locations of the smartphone. In their work, 10 test subjects were included in two experiments to perform 6 daily activities: one with the phone fixed on the waist in 16 different orientations and another with shirt-pocket, trouser-pocket and waist in two different orientations. They constructed a feature vector calculating the mean and standard deviation over 3D acceleration signals, and established an instance-based classifier ($k = 3$). By rectifying the acceleration signals into the same coordinate system, the recognition accuracy has been improved significantly. For a set of location-specific classification models, the accuracy could be up to about 90%.

Sun *et al.* [26] used an accelerometer to distinguish 7 physical activities conducted by people every day. Besides, this paper specified 6 pocket positions. Compared with former two studies, this paper construed more features from data collected from 7 subjects, including not only time-domain features but also frequency-domain features, and then it implemented SVM classifier. With known pocket position, the overall F-score could reach 94.8%.

Kwapisz *et al.* [27] collected accelerometer data from 29 volunteers, conducting 5 daily activities. The phone was carried in subjects' leg pocket of front pants. The features extracted in this paper were statistic values in time domain. Finally, they build up 4 classifiers, and showed a high accuracy over 90% for most activities.

We can see that the accelerometer has received the most attention in previous human activity recognition study. However, in recent years, other sensors like gyroscope have attracted researchers' focus. Wu *et al.* [28] proposed an activity recognition approach using accelerometer and gyroscope data for 16 participants on 13 activities. Seven classifiers had been implemented and the result showed that the accuracy was higher than solely on accelerometer. Shoaib *et al.* [29] made a comprehensive study on impact of position to carry the phone, combination of sensors and the performance for different classifiers, and they built up generalized model and personalized model to investigate the influence of user space.

These previous studies have shown that HAR is a complex task, where many factors would all affect the accuracy of HAR, and their approach frameworks and evaluation procedures are different. To our knowledge, there are quite a few systematic studies like [29] for smartphone-sensor based activity recognition, and a few researches present a systematic performance evaluation in this field. Thus, in our study, we try to give a detailed activity recognition approach, from data acquisition, data preprocessing, feature construction, classifier implementation to activity discrimination. Moreover, we make a full-scale performance evaluation with extensive results, with the hope that our conclusions can contribute to improve the activity recognition technology.

TABLE 2. Brief summary of previous work.

Source Study	Devices	Sensors[2]	Classification Method	Features[3]	Accuracy[4]	Subjects	Activities	Placements
Tomas *et al.* [24]	Nokia N95	Acc.	k-NN	T.	90%	N[1]	6	Fixed position
Apiwat *et al.* [25]	iPhone	Acc.	Instance-based	T.	91.71%	10	6	2 (pocket) & 16 (waist)
Lin *et al.* [26]	Nokia N97	Acc.	SVM	T. & F.	94.8%[5]	7	7	6
Jennifer *et al.* [27]	Android phone	Acc.	4 classifiers	T.	91.7%	29	5	Front pants leg pocket
Wanmin *et al.* [28]	iPod Touch	Acc. & Gyr.	7 classifiers	T. & F.	94.1%[6]	16	13	Fixed position
Muhanmmad *et al.* [29]	Samsung Galaxy SII	Acc. & Gyr. & Mag.	9 classifiers	T. & F.	About 98%	10	5	5

[1] "N" means this item is not mentioned in the literature.
[2] For clarity, Acc. refers to Accelerometer, Gyr. refers to Gyroscope, and Mag. Refers to Magnetometer.
[3] For clarity, T. refers to time-domain, and F. refers to frequency-domain.
[4] Since each literature includes various research objects, here we give their best results.
[5] In [26], the accuracy is given in the form of F-score.
[6] In [28], the highest accuracy is 100 % for sitting. Here we give the second best result (for walking on a level ground).

III. DATA ACQUISITION AND PREPROCESSING

In this part, we offer considerable details regarding the conduct of data collection and preprocessing.

TABLE 3. Body shape information of subjects.

Weight (kg) \ Height (cm)	[165, 170)	[170, 175)	[175, 180)
[50, 60)	Subject 4,5	Subject 9	Subject 6, 10
[60, 70)	-	Subject 2, 7	Subject 1, 6, 8
[70, 80)	-	Subject 3	-

A. DATA COLLECTION

In our work, we built up a data collection application running on MEXZU MX3 with Android OS 4.4.x. The smartphone is equipped with 3 axis-accelerometer, gyroscope and magnetic field sensor. When activated, the application continuously reads linear accelerometer and gyroscope sensor data from three axes of each sensor and timestamp at about 20Hz (the actual sample rate will slightly fluctuate around the setting value during the data collection procedure), and then the data are stored into a local plaintext file. In the evaluation phase, the raw data are transferred to a backend server.

We recruited 10 subjects in the university campus, whose heights range from 165cm to 178cm and weights range from 55kg to 72kg. All 10 subjects are male, aging from 20 to 23. In view of the request for privacy protection from subjects, we only show the approximate range of their height and weight in Table 3. In our experiment, each subject was required to conduct twenty-five rounds of data collection (five human activities × five smartphone placement settings). To make our experimental results typical and representative, in our experimental task, we chose five common actions in our daily life: descending stairs, ascending stairs, walking, jogging and jumping. Furthermore, we find trousers pocket, jacket pocket, hand and upper arm are the four main positions people use to carrying smartphones. Besides, since the waist is close to human body's mass center, waist-placement for motion sensors is commonly used in many studies [30]. Thus, in this study, we got five smartphone placement settings: right upper arm, right hand, right jacket pocket, right trousers pocket and waist.

In our experiment settings, the jogging/walking distance was about 200m, and the descending/ascending height was between 1st floor and 8th floor. To ensure each subject's movements were natural, we did not control the speed and range of their movements. On the contrary, we told them to perform these actions at ease, and between two rounds, they would take a break. Considering the fact that jumping is really one kind of tiring action and people rarely jump in their daily life, for every placement setting, each subject only needed to jump more than fifteen times. Hence, as a result, the size of jumping dataset was far too small compared with other four actions. In section VI, we will discuss this problem in detail.

B. DE-NOISING

Generally, raw sensor data are always noise-corrupted, which leads to measurement inaccuracies and makes it hard to reflect the motion changing of smartphones accurately. To mitigate the effect of noise in obtained data, filtering technique needs to be employed. The most common filtering methods include mean filter, low-pass filter, Gaussian filter, wavelet filter and Kalman filter, etc. After considering the trade-off between computing complexity and de-noising performance, here we employ cubical smoothing algorithm with five-point approximation as follows:

$$\begin{cases} \bar{y}_{-2} = (69y_{-2} + 4y_{-1} - 6y_0 + 4y_1 - y_2)/70 \\ \bar{y}_{-1} = (2y_{-2} + 27y_{-1} + 12y_0 + 4y_1 - y_2)/35 \\ \bar{y}_0 = (-3y_{-2} + 12y_{-1} + 17y_0 + 12y_1 - 3y_2)/35 \\ \bar{y}_1 = (2y_{-2} - 8y_{-1} + 12y_0 + 27y_1 + 2y_2)/35 \\ \bar{y}_2 = (-y_{-2} + 4y_{-1} - 6y_0 + 4y_1 - 69y_2)/70 \end{cases} \quad (1)$$

where $(y_{-2}, y_{-1}, y_0, y_1, y_2)$ are five adjacent points of a data series, while $(\overline{y_{-2}}, \overline{y_{-1}}, \overline{y_0}, \overline{y_1}, \overline{y_2})$ are the filtered samples.

Compared with Kalman filter, the de-noise performance of this filter method is not good enough. However, the computing procedure is more simple and faster, meaning a lighter computational load for smartphones.

C. DATA SEGMENTATION

1) IMPLEMENTATION OF TIME WINDOW

Though most noise gets mitigated, the filtered data are still continuous and long-lasting, so we cannot directly use it as training or testing data. To solve this problem, an overlapped time window is applied, which segments the filtered data into many short ones. In our work, the length of each window is set 1 second, while the degree of overlapping is set as 50%.

2) ACTION SEGMENTATION

In our recognition approach, we draw every single action from the action series as the basic unit for feature extraction. That is, we also need an action segmentation method to the data sequence in a time window, for splitting the continuous data into single action segments. In some previous literatures related to giant recognition or step detection, this procedure, called "cycle detection", has been widely discussed [39], [40]. The main idea of cycle detection is to find specific points in time-acceleration series, like zero-cross points, peak points and valley points. The rationale is intuitive: every action unit follows an accelerate-decelerate pattern, so in an ideal condition, the value of accelerometer of one axis will undergo a "zero → positive value → zero → negative value → zeros" procedure (or "zero → negative value → zero → positive value → zeros", depending on the coordinate system). However, due to the tremble occurring sensor events during the data collection procedure, simply finding specific points, e.g. zero-cross points, may lead to high false positive rate. As a result, we would get much more action segments than the actual number. Kang and Han [1] proposed a novel step detection algorithm that detects each step by peak value and can ignore some pseudo peak points. In this work, we employ this step detection technique to help us implement the action segmentation:

$$t^{peak} = \left\{t | a_t^{peak} > a_{t+i}^{peak}, a_t^{step} > a_\tau^{step}, |t| \le N/2, i \ne 0\right\} \tag{2}$$

$$t^{pp} = \left\{t | \max(a_t^{peak} - a_{t+i}^{peak}) > a_\tau^{pp}, 1 \le |t| \le N/2\right\} \tag{3}$$

$$t^{slope} = \left\{t | \frac{2}{N}\sum_{i=t-N/2}^{t-1}(a_{i+1}^{step} - a_i^{step}) > 0, \frac{2}{N}\sum_{i=t+1}^{t+N/2}(a_i^{step} - a_{i-1}^{step}) < 0\right\} \tag{4}$$

$$t^{step} = t^{step} \cap t^{pp} \cap t^{slope} \tag{5}$$

$$(t^{start}, t^{end}) = \left\{(t_i^{step}, t_{i+1}^{step}) | t_{i+1}^{step} - t_i^{step} < \tau\right\} \tag{6}$$

where $\left\{a_1^{step}, a_2^{step}, \ldots, a_t^{step}\right\}$ is the linear acceleration (z-axis) sequence in a time window, t_i^{step} is the step segmentation point, and between (t^{start}, t^{end}) is the sequence of an action unit. Here we set $a_\tau^{pp} = 0.3\text{m/s}^2$, $N = 10$, and $\tau = 1.2\text{s}$.

IV. FEATURE EXTRACTION

A. SENSOR-BEHAVIOR FEATURES

1) *Time-domain features*

Mean: The direct component of the signal.

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{7}$$

Standard deviation: Inflect the scatter of the data. In the other words, the standard deviation can show the intensity of people's activities.

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})^2} \tag{8}$$

Max and Min: Show the changing range of the signal.

Correlation: Represent the correlation between each pair of axes of the sensor data.

$$Corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{9}$$

$$cov(X, Y) = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y}) \tag{10}$$

Interquartile range: The difference between the third and first quartiles in descriptive statistics, which can also inflect the changing range of the signal.

DTW (Dynamic time warping distance): DTW is widely applied in the field of speech recognition to cope different speaking speeds. Generally speaking, DTW is aimed to measure similarity between two temporal sequences varying in time. Features mentioned previous are statistical, which can't show the essential difference between two different subjects' actions. To solve this problem, we select a standard template for each action and then calculate the DTW distance between the tested data and the template as an important feature.

2) *Frequency-domain features:* ***FFT coefficients***

Since human actions are in low-frequency domain and the calculation of FFT is time-consuming, this paper adopted the second to ninth FFT coefficients as frequency-domain features (Because the first FFT coefficient represents the direct component, which is the same with mean value of the sequence, so we do not use the first FFT coefficient.).

3) *Wavelet-domain features:* ***wavelet energy***

A wavelet is a mathematical function used to divide a given function or continuous-time signal into different scale components. Since wavelet transform has advantages over traditional Fourier transforms

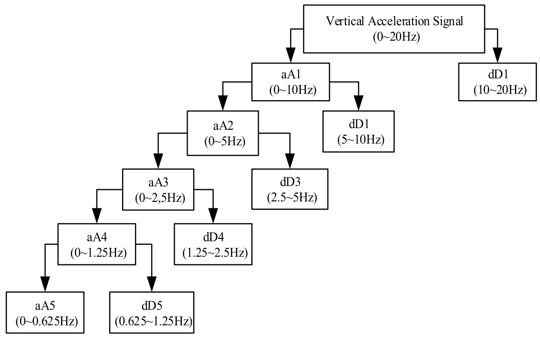

FIGURE 1. Wavelet decomposition process.

for representing functions in different scale components, it successfully combines the time-domain and frequency-domain information. Gradually, it gains more and more attention in the researches on human activity recognition and action-based identification. Wang *et al.* [19] used the fifth-order Daubechies wavelet in the wavelet decomposition applied to the acceleration signal with decomposition at six levels. Xue and Jin [9] adopted wavelet energy of acceleration data in vertical direction to improve the recognition accuracy between going upstairs and downstairs.

In this paper, the sampling rate of the sensor was about 20Hz, here we applied three-order Daubechies wavelet at five levels to decompose the acceleration data in vertical direction (shown in Fig. 1), and extract detailed coefficients at four and five levels to calculate the wavelet energy, which approximated the 0.625∼2.5Hz signal band.

$$E_{wavelet} = \|dD_4\|^2 + \|dD_5\|^2 \tag{11}$$

B. FEATURE SELECTION

After feature extraction, we got an 89-dimensional feature vector (as shown in Table 4). To accelerate the process of computation and compress data, it is necessary to do dimensionality reduction. Furthermore, not all these features we extracted is "good" enough, and the "no good" features should be eliminated. Here we perform feature analysis and select the effective features to speed up the computation as well as reduce the memory used.

We conduct two-sample Kolmogorov-Smirnov test (K-S test) to test if two feature datasets are significantly different. K-S test is a nonparametric statistical hypothesis test based on maximum distance between the empirical cumulative distribution functions of the two data sets. In our evaluation, the two hypotheses of K-S test are:

H_0: *the two datasets are from the same distribution.*
H_1: *the two datasets are from different distributions.*

The K-S test reports a *p*-value parameter which measures the level of support for the original hypothesis. The *p*-value parameter represents the occurrence probability of the samples' result when H_0 is assumed to be true. If this *p*-value is smaller than a significant level α, we will reject H_0 hypothesis because events with small probabilities happen rarely.

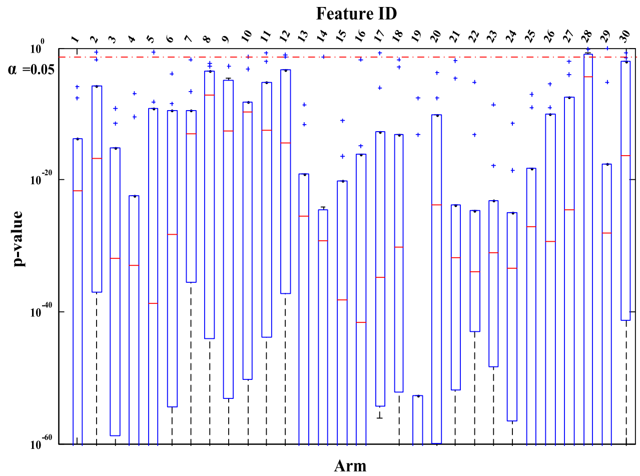

FIGURE 2. K-S test result of 1st∼30th features for arm-placement setting.

For different placement settings, we perform K-S test to all the extracted features pairs, respectively. Fig. 2. Shows the K-S test results of different placements. The top and the bottom of the blue rectangular respectively denote the upper quartile and lower quartile of *p*-values, the red line segment denotes the median value of *p*-value, and the little blue crosses represent the outliers. The red dashed line in the figure denotes the significant level $\alpha = 0.05$.

Due to space limitation, here we only take the arm-placement setting as the example to show how we perform feature election, and other experiments follow the same procedure.

Step 1: Calculate p-values for each pair of different activities.
Step 2: Set the significant level α(usually set to 0.05).
Step 3: For each feature, if the maximum *p*-value of this feature is smaller than α, reserve it; otherwise, discard this feature.

After the three steps, we can get the extracted feature ids : 1, 3, 4, 6, 7, 8, 9, 10, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 32, 33, 34, 35, 36, 37, 38, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 60, 61, 63, 64, 65, 67, 68, 71, 72, 73, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 89.

C. FEATURE NORMALIZATION

Feature normalization is usually preferred after the feature selection process, which rescales the features into the same scale. In this paper, the extracted features come from different sensors, and even if the features come from the same sensor, the calculating processes are also different (e.g., mean value and standard deviation value). So, as result, the range of these features' values varies widely, and the classification result may be governed by the features which have a broad range of values. To avoid this problem, we apply z-score normalization to the raw features we extract:

$$\mathbf{z} = (\mathbf{x} - \boldsymbol{\mu})/\sigma \tag{12}$$

Where $\mathbf{x}x$ is the original feature vector, $\boldsymbol{\mu}$ is the mean vector of training vectors, σ is the standard deviation of each training vectors, and $\mathbf{z}$ is the standard scores after normalization.

V. CLASSIFIER IMPLEMENTATION

This section briefly introduces the multi-class classifiers we used. Each classifier discriminates different human activities by analyzing sensor-behavior data.

A. MULTICLASS CLASSIFIER OVERVIEW

Generally speaking, human activity recognition is a typical multi-label classification problem (i.e., more than two activities are included in the classification task). However, most commonly used classifiers are fundamentally binary classifiers (e.g., Linear Discriminant Analysis, Linear Support Vector Machines.). For the traditional two-class classifiers, the common solutions for multi-class task are *one-versus-one* (for K classes, the classifier is trained between each pair of classes, so finally there are K(K-1)/2 classifiers) and *one-versus-rest* (for each of K classes, in the training stage, this class is labeled positive and the others are labeled negative, so finally there are K classifiers) and *many-versus-many* (where a specific coding technique needs to be employed, like ECOC [36]). In this paper, to discriminate the five kinds of activities, we implemented three multi-class classifiers: Nearest Neighbors, Random Forests and Support Vector Machines.

B. CLASSIFIER 1: NEAREST NEIGHBOR

The Nearest-Neighbor classifier models one activity's motion-sensor behavior on the assumption that new feature vectors from the same activity will resemble one or more of vectors in the training data. In our work, we implemented a K-Nearest-Neighbor classifier, which assigns the new vector the label most frequently represented among the k nearest training samples [34]. In the training phase, the classifier calculates the covariance matrix of the training feature vectors, and the nearest-neighbor parameter k is set as 5. In the testing phase, the classifier computes Euclidean distances, and the new sample is assigned to the category most frequently appearing among the k nearest training samples.

C. CLASSIFIER 2: RANDOM FORESTS

Random Forests is an ensemble method using tree-type classifiers for classification, which consists of a number of trees grown from bootstrapped sets of the original training data. One of the most prominent advantages of Random Forests is that Random Forests can handle data with high dimensionality by increasing the number of trees, which is very suitable for the feature vectors extracted in former section. In the training phase, a number of bootstrapped sets are generated from the original raining samples, and then N trees are grown, in each of which every node is split by a random selection of predictor features. To obtain a high accuracy and lighter computing load, here we set the number of trees N as 200. In the testing phase, a test sample is put into the classifiers, and then every tree will give an output. The sample is classified by a majority vote of the outputs.

D. CLASSIFIER 3: SUPPORT VECTOR MACHINES

The core idea of SVM (Support Vector Machines) is to map training examples marks as one of two categories into a high space, and then find the best hyperplane to separate the two categories. It should be noted that SVM is fundamentally a two-class classifier [35], so we need to apply one multi-classification approach to traditional SVM. Since *one-versus-rest* strategy will lead to inconsistent results and harm the symmetry of the original problem [35], here we choose LibSVM [34] which implements the *one-versus-one* approach in this work. In the training phase, the classifier is established by using the training vectors with the linear kernel function and radial basis kernel function (RBF), and we use the default parameter. In the testing phase, the classifier projects the test vector onto the same high-dimensional space, and classifies it into the category with the highest score.

VI. PERFORMANCE EVALUATION METHODOLOGY

This section depicts how we evaluate the performance of the activity recognition approach proposed in this paper.

A. TRAINING AND TESTING PROCEDURE

To make a comprehensive investigation of the impact of the user space, we consider three conditions in this paper: 1) personalized model: the training data and the testing data are all from the same subject (*one-to-one model*); 2) generalized model 1: the training data come from all the subjects, and the testing data come from only one subject (*all-to-one model*); 3) generalized model 2: the testing data are from only one subject and the training data are from the rest subjects (*rest-to-one model*).

1) *Personalized model*

   On-to-one model: Weight, height, gender and physical condition will affect people's movement. In other words, people's activity carries their own personalized pattern. So it is natural to build up a classifier trained only from single user's activity data. For each subject, in the training phase, we only use his training data to get his personalized classification model, and then we use his testing samples to conduct the evaluation.

2) *Generalized model*

   Generally speaking, for a multi-class classification problem, a large amount of training data is needed, especially when the dimensionality of the feature vector is high. However, for activity recognition task, the procedure of collecting training data is tiring. As a result, in most conditions, the training data from a single person is not adequate. Thus, we may have to import other users' training data in practice. Moreover, generalized model is expected to be more robust,

TABLE 4. Motion sensor features representation.

Attribute	Description	Feature Identifier
Mean	Mean of sample $(a_x, a_y, a_z, g_x, g_x, g_z)$ in one action segment.	1, 13, 25, 37, 49, 61
Std	Standard deviation of sample $(a_x, a_y, a_z, g_x, g_x, g_z)$ in one action segment.	2, 14, 26, 38, 50, 62
IQR	Interquartile range of sample $(a_x, a_y, a_z, g_x, g_x, g_z)$ in one action segment.	3, 15, 27, 39, 51, 63
Wavelet Energy	Wavelet energy of sample $(a_x, a_y, a_z, g_x, g_x, g_z)$ in one action segment.	4, 16, 28, 40, 52, 64
FFT	First 2nd ~ 9th magnitude of FFT coefficients of sample $(a_x, a_y, a_z, g_x, g_x, g_z)$ in one action segment.	5~12, 17~24, 29~36, 41~48, 53~60, 65~72
DTW Distance	DTW distance from the template of descending stairs, ascending stairs, walking, jogging and jumping.	73~77
Corr	Correlation of $(a_x, a_y), (a_y, a_z), (a_z, a_x), (g_x, g_y), (g_y, g_z), (g_z, g_x)$	78~83
Max - Min	The difference between the maximal value and minimal value of samples in one action segment.	84~89

which can reduce abnormal data interference. That is the motivation why we build up the generalized model.

All-to-one model: As mentioned before, we need the model contains personalized information while we need adequate data and a robust-behavior model. Therefore, we build up the all-to-one model. In the training phase, we use all training data including the tested subject's training data.

Rest-to-one model: The only difference compared with all-to-one model is that in rest-to-one model, the training data exclude the tested subject's training data. We build this model out of two main purposes: to investigate the impact of the personalized information; to examine whether the generalized information can accomplish the activity recognition task with an acceptable accuracy. If so, the activity recognition system can classify one person's activity with a built-in classifier pre-trained by existing training data, which will enhance the system's utility.

B. TRAINING AND TESTING PROCEDURE

To evaluate the recognition approach performance, at first we need to construct training dataset and testing dataset by splitting the whole collected samples. We employ 10-fold cross validation to evaluate the performance [26], [28], [29]. In the validation process, 1) for *one-to-one model*, we randomly divide the data from one single subject into 10 parts with the same size, then we set each subset as the testing data and set the remaining subsets as the training data, and finally we compute the average accuracy as the final result; 2) as for *all-to-one model*, we divide each subjects' data into 10 subsets, and then for i^{th} ($i = 1, 2, \ldots, 10$), we select each subjects' ith subset to compose the testing set, and the remaining compose the training set; 3) for *rest-to-one model*, the data parsing process is the same with *all-to-one model*, while the only difference is the raining set excludes i^{th} subject's own data. It should be noted there may be a slight difference between the class-ratio of the whole dataset and that of the divided subset. To reduce the impact of data imbalance, we calculate the F-score of each activity, and finally we compute the average F-score as the criterion.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents a systematical analysis and evaluation of human activity recognition using smartphone motion-sensor data. In the evaluation procedure, we compare the performance when changing some factors: 1) phone-placement settings, 2) user space, 3) combination of sensors, and 4) degree of data imbalance.

A. ACTION RECOGNITION UNDER VARIOUS PLACEMENT SETTINGS

1) METHOD

The placement setting refers to the position where the smartphone is placed. There are two main aspects how the smartphone's position affects the performance. First, data from smartphone sensors are in the smartphone coordinate system, while people's movements are always measured in the geographic coordinate system. Hence, the sensor data is corresponded with the orientation of the smartphone and the geographic coordinate system. Second, different parts of the body show different patterns (e.g., when we are walking, our arms swing widely, while our waists wiggle slightly). Third, in smartphone-sensor-based activity recognition, the smartphone can only monitor the movement at sole position. Hence, it's necessary to study the recognition accuracy under various placement settings. Here, we calculate the accuracy in six placement settings: 1) *arm* (upper right arm), 2) *hand* (right hand), 3) pocket1 (right jacket pocket), 4) *pocket2* (right trousers pocket), 5) *waist* and 6) *no-position-information* (we mix the five positions data together).

2) RESULTS AND ANALYSIS

Here we present the personalized model to show the recognition accuracy in different placement settings. Due to the space limit, here we only show ten subjects' recognition F-score in Fig. 3, and we provide the detailed results in the appendix (TABLE 5 $\sim$ TABLE 8).

The performance for *pocket2* is the best (the accuracy is high and robust for every subject.), while for the hand-hold condition, the performance is the worst. We conjecture that the smartphone placement has significant influence on recognition accuracy. Data collected from smartphone sensors reflect the movement patterns of the body part which

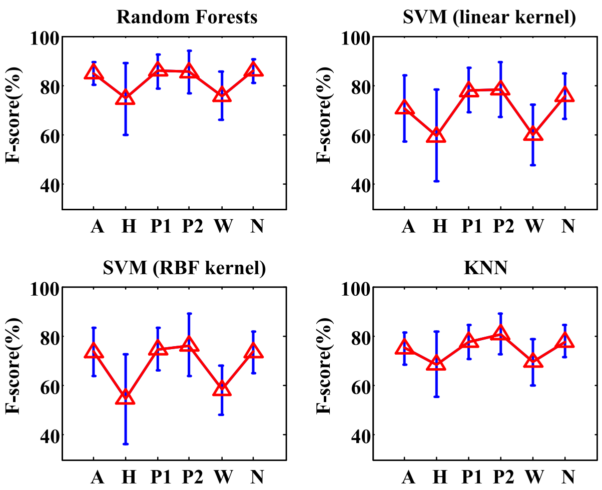

FIGURE 3. Personalized model F-score for various placement settings. The x-axis labels represent the placement settings: "A" – upper arm, "H" – hand, "P1" – jacket pocket, "P2" – trousers pocket, "W" – waist, "N" – no position information.

TABLE 5. Random forests (one-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	79.52%	59.99%	87.83%	82.22%	86.73%	83.59%
User 2	89.92%	47.31%	90.61%	93.10%	83.13%	86.37%
User 3	86.69%	76.12%	70.27%	74.43%	64.75%	84.14%
User 4	83.45%	58.80%	82.99%	65.66%	65.87%	78.26%
User 5	89.49%	89.66%	87.54%	92.05%	71.19%	89.16%
User 6	78.08%	83.87%	78.58%	85.17%	64.27%	77.90%
User 7	85.51%	82.09%	86.43%	92.41%	70.93%	89.17%
User 8	87.74%	80.95%	90.40%	87.84%	81.74%	87.65%
User 9	89.94%	83.70%	93.83%	91.44%	91.96%	91.04%
User 10	79.76%	85.86%	90.02%	86.58%	76.86%	90.44%

TABLE 6. SVM of linear kernel (one-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	61.30%	45.42%	83.57%	79.06%	66.68%	69.21%
User 2	85.10%	38.98%	82.59%	89.14%	50.35%	75.89%
User 3	80.60%	40.50%	66.15%	60.84%	43.62%	73.59%
User 4	66.25%	57.46%	74.33%	64.31%	55.62%	61.11%
User 5	84.56%	89.37%	77.80%	86.80%	56.33%	84.99%
User 6	64.40%	65.33%	72.99%	63.35%	50.80%	62.64%
User 7	65.01%	38.70%	62.43%	87.78%	53.77%	78.98%
User 8	81.66%	69.85%	85.64%	85.61%	74.61%	80.88%
User 9	75.46%	63.57%	91.22%	89.11%	84.28%	85.55%
User 10	42.53%	86.21%	81.79%	76.61%	61.36%	84.32%

the phone is attached to. When the phone is placed in the trousers pocket, the smartphone records how the thigh moves; when the phone is held in hand, the smartphone records how the hand moves. However, the movement patterns of hand are more complex: for instance, when someone walks, his hands swing not only in anterior-posterior direction but also in left-right direction. If we take the movement of wrists into consideration, the hands' movement pattern will be extremely complex. The movement patterns of thigh are simpler

TABLE 7. SVM of RBF kernel (one-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	63.10%	44.02%	84.04%	76.91%	66.94%	69.15%
User 2	85.31%	37.69%	82.26%	88.32%	53.57%	75.99%
User 3	80.15%	41.29%	64.90%	59.83%	42.76%	73.00%
User 4	65.21%	50.06%	72.83%	66.23%	60.99%	61.36%
User 5	80.64%	87.94%	74.91%	86.20%	56.06%	85.00%
User 6	65.83%	66.77%	72.57%	60.14%	51.00%	63.07%
User 7	64.29%	39.30%	61.77%	88.70%	59.70%	78.82%
User 8	83.88%	69.41%	85.08%	85.00%	74.73%	81.42%
User 9	75.73%	63.10%	92.99%	87.59%	84.32%	85.29%
User 10	42.15%	85.84%	82.30%	74.64%	61.55%	83.96%

TABLE 8. K-nearest neighbor (one-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	67.61%	52.34%	83.74%	78.41%	79.12%	74.93%
User 2	82.53%	49.08%	82.01%	88.07%	82.01%	79.79%
User 3	76.32%	65.56%	62.96%	72.25%	61.05%	73.34%
User 4	74.73%	54.67%	72.58%	63.62%	57.22%	67.36%
User 5	80.44%	85.26%	81.11%	85.56%	62.79%	85.15%
User 6	62.56%	69.29%	73.19%	77.35%	64.98%	70.02%
User 7	70.63%	75.73%	72.15%	87.62%	66.67%	78.82%
User 8	80.78%	66.79%	82.84%	85.20%	72.93%	79.54%
User 9	78.13%	82.72%	84.07%	88.55%	83.16%	86.98%
User 10	76.01%	82.86%	80.74%	81.83%	63.98%	83.11%

by contrast. The thigh mainly moves in anterior-posterior and vertical direction, and its movements are more periodic. What's more, for the five types of activities defined in this work, maybe thigh is the best part to show the discriminative features of them: for walking, jogging and jumping, the force and frequency of thigh's movements are different; as for descending and ascending stairs, the latter needs to raise legs higher. Consequently, we can conclude that, for the single-device scenario, the selection of the position to carry the device is necessary, and we suggest that that position should be chosen depending on the specific task. For example, if you want to discriminate whether a person is opening the door, maybe it's not appropriate to place the smartphone in trousers pocket. In addition, Random Forests shows the best performance compared with other three classifiers, because of its ability of performing feature selection automatically and having fewer parameters to set.

B. SENSITIVITY TO USER SPACE

1) METHOD

According to kinesiology, the range and speed of one people's motion are corresponded with his age, gender, habitus, health condition, strength, and etc. For example, compared with the youth, the elder's stride frequency is lower and step length is shorter; people who keep taking exercises jump higher than those who rarely do sports; an adult's step length is longer than a child's. Of course, the aforementioned conclusions are not always correct, but we want to show that different people tend to behave differently for the same kind of movement.

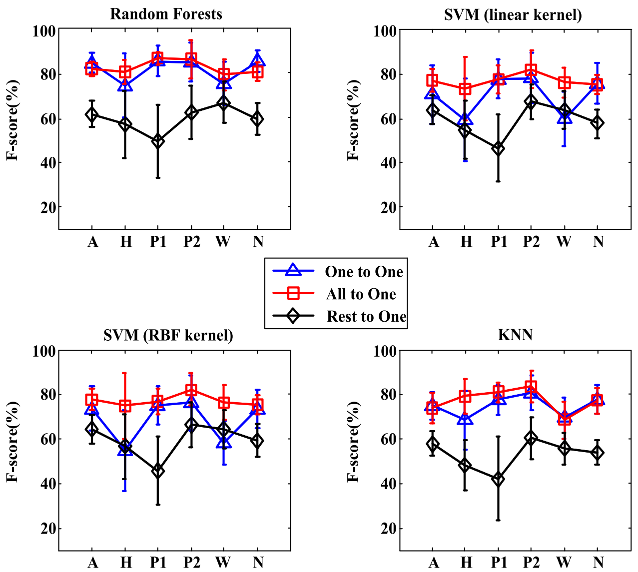

FIGURE 4. F-score in different user spaces. The x-axis labels represent the placement settings: "A" – upper arm, "H" – hand, "P1" – jacket pocket, "P2" – trousers pocket, "W" – waist, "N" – no position information.

Since our approach is a supervised learning task, the completeness of training data will affect the classifier's performance. In this part, we build up three training-test scenarios to investigate our approach's sensitivity to the user space (see more details in section VII.B: *one-to-one*, *all-to-one* and *rest-to-one*).

TABLE 9. Random forests (all-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	67.61%	52.34%	83.74%	78.41%	79.12%	74.93%
User 2	82.53%	49.08%	82.01%	88.07%	82.01%	79.79%
User 3	76.32%	65.56%	62.96%	72.25%	61.05%	73.34%
User 4	74.73%	54.67%	72.58%	63.62%	57.22%	67.36%
User 5	80.44%	85.26%	81.11%	85.56%	62.79%	85.15%
User 6	62.56%	69.29%	73.19%	77.35%	64.98%	70.02%
User 7	70.63%	75.73%	72.15%	87.62%	66.67%	78.82%
User 8	80.78%	66.79%	82.84%	85.20%	72.93%	79.54%
User 9	78.13%	82.72%	84.07%	88.55%	83.16%	86.98%
User 10	76.01%	82.86%	80.74%	81.83%	63.98%	83.11%

2) RESULTS AND ANALYSIS

Here we present the results of generalized model and personalized model in the same figure to compare their performance. The result is shown in Fig. 4, and detailed results can be seen in the appendix (TABLE 9 ∼ TABLE 16).

Comparing *one-to-one model* with *rest-to-one model*, we can see obvious performance degradation in *rest-to-one model*. The reason is apparent: each person has his own biometric characteristics, and even for the same activity, different people show distinct patterns. Hence, for human activity recognition task, it's better to include more personalized data. While for *all-to-one model*, we can observe a significant improvement of both accuracy and stability. There may be two reasons: first, for *all-to-one model*, it has more training data, including personalized data; second, personalized data may contain some abnormal samples, which may be caused

TABLE 10. SVM of linear kernel (all-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	77.65%	57.53%	83.42%	84.65%	77.48%	76.97%
User 2	87.21%	41.83%	85.20%	88.21%	80.05%	74.83%
User 3	82.76%	77.31%	79.74%	76.18%	67.86%	74.26%
User 4	75.05%	65.80%	77.67%	64.20%	68.46%	65.33%
User 5	76.66%	80.65%	72.49%	93.13%	75.01%	76.29%
User 6	69.64%	78.75%	78.20%	77.80%	68.42%	73.05%
User 7	76.39%	86.25%	75.94%	87.28%	81.00%	79.80%
User 8	80.53%	86.87%	79.99%	84.02%	73.45%	77.11%
User 9	72.41%	83.81%	62.86%	88.37%	85.31%	80.56%
User 10	72.26%	74.94%	81.15%	77.05%	85.10%	74.64%

TABLE 11. SVM of RBF kernel (all-to-one model)

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	77.95%	58.25%	81.62%	84.83%	76.56%	77.03%
User 2	87.67%	40.92%	85.83%	87.66%	84.05%	75.03%
User 3	82.71%	79.36%	78.87%	78.07%	72.15%	73.79%
User 4	77.97%	68.45%	76.33%	65.61%	62.03%	65.74%
User 5	76.20%	82.65%	71.55%	90.16%	73.68%	76.55%
User 6	71.87%	80.08%	74.75%	77.44%	69.79%	73.26%
User 7	74.56%	85.02%	77.28%	88.78%	81.80%	80.00%
User 8	81.04%	86.44%	78.40%	83.23%	72.89%	77.25%
User 9	73.72%	84.56%	63.84%	88.07%	86.92%	80.70%
User 10	73.47%	81.77%	79.38%	78.64%	85.42%	74.19%

TABLE 12. K-nearest neighbor (all-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	72.76%	69.64%	81.53%	84.77%	68.10%	76.91%
User 2	82.79%	72.79%	85.63%	92.66%	80.22%	83.06%
User 3	70.21%	71.04%	76.57%	73.76%	67.38%	72.41%
User 4	69.39%	68.83%	70.98%	71.47%	60.34%	65.44%
User 5	77.31%	83.93%	81.71%	88.75%	64.63%	80.11%
User 6	59.75%	80.61%	83.40%	80.45%	53.33%	71.27%
User 7	77.03%	86.35%	77.91%	87.54%	67.84%	79.48%
User 8	74.82%	86.85%	86.06%	86.05%	67.20%	79.77%
User 9	74.52%	85.41%	83.69%	89.82%	77.35%	83.40%
User 10	81.38%	86.25%	82.23%	79.63%	78.44%	81.65%

TABLE 13. Random forests(rest-to-one model).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	60.58%	52.87%	74.30%	67.76%	66.76%	69.26%
User 2	65.94%	43.02%	41.02	53.16%	62.12%	57.77%
User 3	59.66%	53.51%	66.39	59.89%	74.31%	62.20%
User 4	58.94%	59.96%	68.59	53.90%	52.34%	55.91%
User 5	59.29%	64.76%	21.36	43.12%	59.83%	61.51%
User 6	63.21%	74.71%	57.64	68.68%	59.89%	65.79%
User 7	52.53%	46.07%	45.65	75.60%	75.43%	63.71%
User 8	74.45%	77.39%	44.55	51.24%	64.12%	54.85%
User 9	67.26%	69.59%	37.06	77.48%	84.70%	63.90%
User 10	57.62%	29.83%	38.59	74.74	72.58%	43.81%

TABLE 14. **SVM of linear kernel (rest-to-one model).**

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	66.85%	53.96%	67.56%	68.15%	66.39%	68.28%
User 2	72.55%	35.11%	37.69%	54.90%	54.66%	55.31%
User 3	68.27%	55.75%	62.36%	63.53%	59.17%	59.69%
User 4	55.79%	57.16%	65.10%	57.86%	54.98%	45.08%
User 5	57.09%	60.85%	29.47%	66.01%	61.33%	58.30%
User 6	62.63%	59.56%	51.41%	68.96%	62.50%	66.52%
User 7	59.18%	52.06%	42.01%	79.02%	66.41%	55.87%
User 8	73.72%	73.22%	24.24%	67.49%	61.20%	55.88%
User 9	65.55%	68.46%	38.49%	78.19%	84.44%	54.60%
User 10	57.24%	30.52%	47.15%	70.69%	67.41%	58.07%

TABLE 15. **SVM of RBF kernel rest-to-one model).**

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	65.92%	53.92%	67.86%	65.90%	66.20%	68.28%
User 2	73.89%	35.23%	36.36%	55.19%	56.97%	55.31%
User 3	68.19%	58.84%	61.63%	65.02%	63.45%	59.69%
User 4	58.00%	57.80%	65.49%	45.71%	53.19%	45.08%
User 5	55.93%	59.02%	30.07%	62.09%	59.40%	58.30%
User 6	61.63%	74.72%	48.42%	69.94%	62.27%	62.06%
User 7	57.08%	51.81%	43.03%	79.82%	69.64%	61.23%
User 8	73.41%	71.24%	23.85%	68.48%	61.49%	62.33%
User 9	67.50%	71.62%	35.55%	76.65%	84.76%	69.01%
User 10	59.73%	31.43%	44.09%	72.08%	67.27%	51.70%

TABLE 16. **K-nearest neighbor (rest-to-one model).**

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	52.36%	48.01%	62.96%	61.64%	55.20%	59.76%
User 2	65.87%	45.66%	41.31%	43.19%	53.59%	53.78%
User 3	57.52%	37.21%	64.20%	58.27%	58.42%	54.06%
User 4	48.45%	46.72%	59.57%	49.82%	52.42%	39.53%
User 5	56.87%	42.33%	26.41%	55.73%	42.80%	52.57%
User 6	53.81%	65.66%	64.94%	66.12%	52.26%	57.74%
User 7	57.16%	39.31%	23.66%	75.36%	52.19%	54.91%
User 8	63.18%	62.69%	20.26%	60.25%	56.52%	55.36%
User 9	62.89%	61.21%	24.62%	68.02%	69.31%	59.58%
User 10	59.86%	32.90%	32.00%	64.49%	64.70%	52.87%

by severe noise interference or deformation of movement. The additional training data from others can help eliminate the influence of aberrant training samples and enhance the activity recognition system's robustness. According to aforementioned analysis, we believe that *all-to-one* has advantages in robustness enhancement and releasing the load of data collection.

C. STABILITY TO COMBINATION OF MOTION SENSORS

1) METHOD

There are two main motion sensors embedded in smartphones: accelerometer that returns acceleration force data for the three coordinate axes, and gyroscope that returns rate of rotation data for the three coordinate axes. Most previous work chose accelerometer as the data source, while only a few use the gyroscope. Human activity is always complex, including rotation and translation, so which motion form can better reflect the movement needs to be explored. In this part, we show the performance for different combination of motion sensors, including using accelerometer information solely, gyroscope information solely, and both accelerometer and gyroscope information. There are some points that should be noted: first, we use the *linear accelerometer* to exclude gravity; second, for single-sensor scenarios, there may be no feature preserved after feature selection, and in this case, we select the features with the minimal *p*-values; third, the performance is evaluated under personalized model; fourth, for single-sensor scenarios, the DTW distance features are excluded, because the original template is built up under two-sensor scenario.

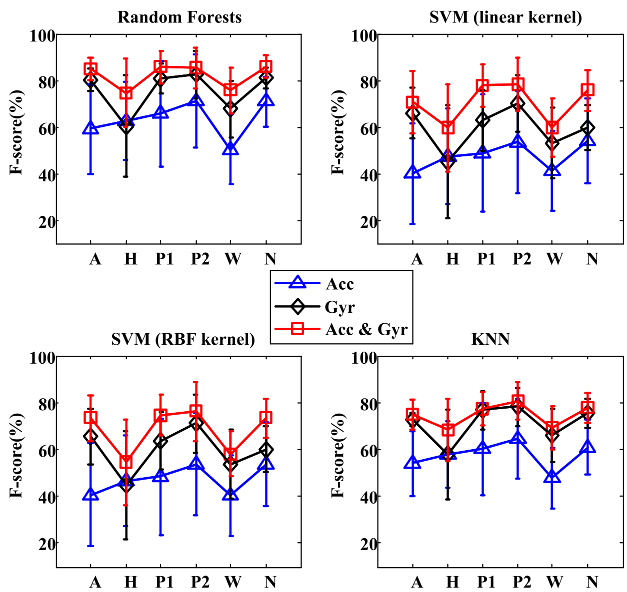

FIGURE 5. **F-score of personalized model for different combinations of sensors. The x-axis labels represent the placement settings: "A" – upper arm, "H" – hand, "P1" – jacket pocket, "P2" – trousers pocket, "W" – waist, "N" – no position information.**

TABLE 17. **Random forests (one-to-one model, acc. only).**

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	36.08%	48.71%	58.49%	70.33%	61.43%	63.72%
User 2	77.09%	27.64%	75.22%	86.05%	54.69%	71.28%
User 3	71.95%	66.70%	22.99%	44.92%	32.48%	62.39%
User 4	54.29%	57.46%	29.61%	27.38%	47.43%	56.53%
User 5	75.88%	83.92%	69.30%	87.16%	44.98%	81.68%
User 6	55.07%	76.76%	72.91%	67.68%	25.77%	56.54%
User 7	44.25%	63.61%	78.04%	79.60%	53.00%	74.10%
User 8	73.59%	78.63%	87.77%	82.16%	46.96%	79.25%
User 9	79.19%	68.66%	88.78%	88.28%	79.61%	85.44%
User 10	24.44%	54.29%	75.55%	79.22%	58.41%	79.94%

2) RESULTS AND ANALYSIS

Figure 5 provides the result for different combinations of sensors (detailed results can be seen in the appendix TABLE 17 ~ TABLE 24). In comparison, we can observe

TABLE 18. SVM of linear kernel (one-to-one model, acc. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	35.69%	45.32%	35.08%	60.99%	45.55%	48.18%
User 2	59.56%	22.42%	57.71%	80.43%	34.77%	57.52%
User 3	53.64%	43.57%	20.74%	19.07%	28.09%	44.78
User 4	24.78%	54.60%	12.45%	20.45%	51.74%	26.51%
User 5	63.35%	85.24%	56.09%	66.80%	31.16%	70.53%
User 6	19.01%	59.99%	53.30%	52.81%	10.03%	24.78%
User 7	13.89%	17.18%	24.38%	49.84%	43.19%	54.93%
User 8	60.82%	68.98%	81.66%	42.74%	46.55%	73.00%
User 9	60.79%	38.01%	87.36%	83.33%	75.64%	76.41%
User 10	10.19%	38.72%	57.91%	59.78%	44.73%	63.82%

TABLE 19. SVM of RBF kernel (one-to-one model, acc. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	35.69%	45.32%	35.08%	60.99%	45.55%	48.18%
User 2	59.56%	22.42%	57.71%	80.43%	34.77%	57.52%
User 3	53.64%	43.57%	20.74%	19.07%	28.09%	44.78
User 4	24.78%	54.60%	12.45%	20.45%	51.74%	26.51%
User 5	63.35%	85.24%	56.09%	66.80%	31.16%	70.53%
User 6	19.01%	59.99%	53.30%	52.81%	10.03%	24.78%
User 7	13.89%	17.18%	24.38%	49.84%	43.19%	54.93%
User 8	60.82%	68.98%	81.66%	42.74%	46.55%	73.00%
User 9	60.79%	38.01%	87.36%	83.33%	75.64%	76.41%
User 10	10.19%	38.72%	57.91%	59.78%	44.73%	63.82%

TABLE 20. K-nearest neighbor (one-to-one model, acc. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	36.71%	45.23%	35.58%	58.71%	45.45%	48.11%
User 2	60.03%	22.58%	55.78%	82.01%	34.36%	57.53%
User 3	52.97%	44.50%	21.05%	19.77%	27.76%	43.20%
User 4	25.33%	53.00%	13.24%	20.62%	47.79%	26.68%
User 5	61.83%	83.10%	56.18%	68.73%	30.46%	70.03%
User 6	19.98%	57.14%	52.76%	53.53%	10.09%	24.48%
User 7	13.55%	18.30%	24.71%	49.60%	38.87%	55.72%
User 8	64.53%	67.10%	81.20%	42.93%	48.67%	72.21%
User 9	62.20%	39.04%	87.68%	83.46%	76.79%	76.87%
User 10	10.28%	37.52%	56.87%	58.66%	44.25%	62.94%

TABLE 21. Random forests (one-to-one model, Gyr. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	79.29%	35.77%	86.43%	82.41%	78.20%	77.62%
User 2	75.71%	27.37%	85.57%	91.40%	78.99%	80.95%
User 3	78.92%	49.86%	67.83%	67.79%	57.41%	79.80%
User 4	79.04%	36.78%	80.29%	61.31%	49.62%	75.35%
User 5	83.97%	86.53%	78.45%	86.57%	58.88%	86.73%
User 6	70.24%	71.97%	73.41%	79.42%	60.67%	73.68%
User 7	85.44%	70.29%	87.63%	91.93%	65.50%	84.92%
User 8	82.39%	60.87%	80.61%	87.60%	77.15%	79.90%
User 9	83.13%	82.36%	86.48%	92.01%	87.88%	87.28%
User 10	85.15%	81.48%	82.19%	83.53%	63.31%	83.36%

that combing two sensors can improve the recognition accuracy and stabilize the performance for different subjects. Moreover, another observation deserves our attention: for two single-sensor scenarios, the gyroscope can give higher

TABLE 22. SVM of linear kernel (one-to-one model, Gyr. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	55.05%	32.00%	81.80%	61.84%	64.68%	52.25%
User 2	66.40%	23.27%	80.10%	65.99%	54.33%	51.03%
User 3	67.73%	25.58%	66.14%	57.71%	39.80%	64.06%
User 4	71.14%	17.54%	72.95%	60.29%	36.10%	52.50%
User 5	80.09%	85.71%	65.10%	82.63%	54.74%	76.65%
User 6	70.08%	25.78%	66.40%	51.47%	50.19%	46.03%
User 7	59.81%	61.59%	45.53%	80.36%	48.21%	67.26%
User 8	76.67%	39.76%	52.70%	84.80%	76.15%	61.93%
User 9	70.62%	63.99%	48.72%	80.81%	75.60%	69.19%
User 10	43.02%	75.35%	49.29%	75.29%	32.72%	58.23%

TABLE 23. SVM of RBF kernel (one-to-one model, Gyr. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	53.50%	32.91%	80.90%	61.66%	65.25%	52.02%
User 2	66.27%	25.39%	81.33%	66.39%	55.90%	51.25%
User 3	71.29%	25.97%	66.39%	58.17%	41.19%	64.35%
User 4	73.55%	17.54%	72.69%	61.16%	35.77%	51.93%
User 5	79.39%	81.25%	63.21%	85.77%	54.36%	76.50%
User 6	67.45%	27.86%	67.00%	51.09%	50.57%	46.47%
User 7	56.21%	59.74%	57.59%	80.75%	48.50%	67.62%
User 8	78.73%	38.18%	51.58%	83.33%	73.36%	62.03%
User 9	69.15%	64.56%	48.35%	82.70%	78.01%	69.71%
User 10	41.85%	75.15%	48.96%	80.14%	36.09%	58.86%

TABLE 24. K-nearest neighbor (one-to-one model, gyr. only).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	71.63%	37.54%	84.18%	75.53%	74.22%	71.75%
User 2	73.99%	31.35%	85.64%	87.97%	78.92%	77.22%
User 3	69.71%	47.49%	61.94%	67.79%	55.01%	71.26%
User 4	73.00%	37.77%	74.83%	62.45%	44.28%	66.49%
User 5	73.33%	83.96%	68.59%	84.40%	63.83%	83.77%
User 6	67.00%	53.90%	66.78%	73.45%	61.49%	69.14%
User 7	75.86%	68.42%	78.81%	80.64%	61.42%	76.46%
User 8	77.20%	60.80%	78.59%	82.91%	75.18%	74.19%
User 9	72.07%	78.09%	84.07%	85.28%	80.90%	85.85%
User 10	74.55%	79.16%	84.61%	83.51%	65.66%	79.14%

accuracy compared with accelerometer, which is not extraordinary since most previous work employed accelerometer as the main sensor. We conjecture that one possible cause is that our proposed approach and research object (e.g., selected activities) differ from other studies. Shoaib *et al.* [29] also finds the top-rank sensors for different activities are not the same. Our results suggest that the choice of sensors should depend on the specific task; on the contrary, it is better to combine multiple types of sensors.

D. IMPACT OF DATA IMBALANCE

1) METHOD

If the classification categories are not approximately equally represented, we call the dataset imbalanced [37], which is common across various disciplines, such as anomaly detection and medical diagnosis. In the imbalanced-data scenario, for the majority class, the classifier always shows a good performance. But for the minority class, the classifier behaves poorly. The main reason is that majority has a more powerful influence in the traditional training stage. Consequently, it is imperative to rebalance the dataset. In general, the rebalancing techniques can mainly be categorized as two ways: oversampling the minority class and undersampling the majority class. In this work, we also faced this problem. For the five kinds of movements in our experiment, jumping has much lower occurrence frequency compared with other four. Here, we use the SMOTE algorithm [38] to oversample the minority class (jumping). The number of K-Nearest Neighbors is set to $k = 3$, and we repeat this oversampling procedure twice. As a result, the jumping class's dataset have increased to threefold.

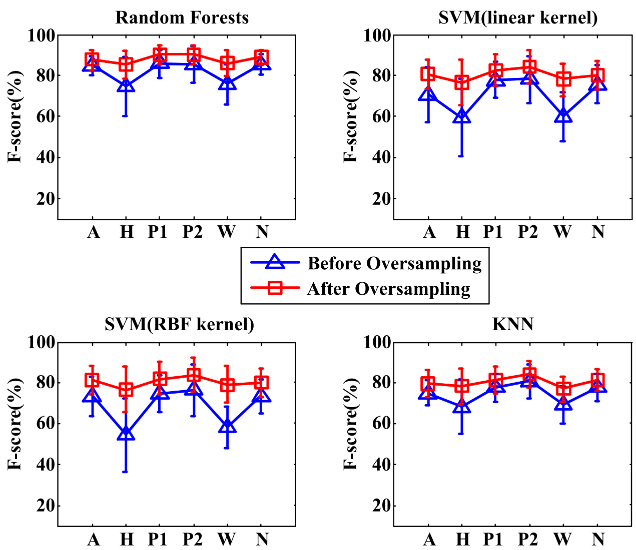

FIGURE 6. Comparison of F-score of personalized model before oversampling and after oversampling. The x-axis labels represent the placement settings: "A" – upper arm, "H" – hand, "P1" – jacket pocket, "P2" – trousers pocket, "W" – waist, "N" – no position information.

TABLE 25. Random forests (one-to-one model, oversampled).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	84.28%	81.35%	92.69%	90.68%	93.43%	87.85%
User 2	94.52%	78.97%	92.20%	95.95%	90.83%	91.34%
User 3	91.65%	87.51%	80.39%	82.93%	91.55%	87.17%
User 4	88.19%	70.99%	83.62%	81.04%	86.33%	83.81%
User 5	87.87%	94.27%	94.04%	93.20%	83.93%	91.62%
User 6	80.39%	85.57%	88.67%	89.12%	74.46%	81.48%
User 7	88.91%	87.57%	90.00%	93.19%	76.82%	91.33%
User 8	89.88%	86.86%	93.48%	92.44%	85.20%	90.06%
User 9	91.08%	89.94%	94.82%	94.24%	93.15%	92.55%
User 10	79.42%	91.00%	92.31%	89.00%	87.16%	91.82%

2) RESULTS AND ANALYSIS

Figure 6 (detailed results can be seen in the appendix TABLE 25 $\sim$ TABLE 28) shows that not only a significant performance improvement of F-score (about 5% in average) after oversampling the jumping class dataset, but also a stabilization of the variance of F-score. In our work, the recognition is a multi-class classification problem with an unequal distribution between five activities. Without resampling technique, the jumping activity, considered as the minority class, will compromise the performance of learning algorithms adopted in this paper.

TABLE 26. SVM of linear kernel (one-to-one model, oversampled).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	72.30%	74.46%	85.17%	83.77%	90.05%	75.81%
User 2	90.46%	58.11%	87.28%	94.74%	82.47%	85.53%
User 3	89.98%	71.36%	73.74%	73.58%	83.53%	78.11%
User 4	78.25%	61.86%	71.63%	72.83%	75.93%	68.01%
User 5	86.85%	94.34%	86.57%	88.88%	72.09%	87.82%
User 6	76.80%	72.56%	82.00%	75.36%	72.10%	68.86%
User 7	75.34%	83.12%	71.07%	91.43%	63.83%	80.25%
User 8	83.49%	83.61%	89.59%	87.05%	74.20%	83.85%
User 9	83.89%	82.22%	93.50%	92.89%	86.32%	86.25%
User 10	69.62%	83.94%	83.68%	81.47%	81.57%	85.14%

TABLE 27. SVM of RBF kernel (one-to-one model, oversampled).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	72.70%	75.34%	85.60%	84.54%	89.93%	76.09%
User 2	90.52%	59.62%	87.10%	94.66%	94.64%	85.80%
User 3	90.61%	70.26%	74.36%	73.91%	82.38%	78.15%
User 4	78.17%	61.46%	72.13%	68.12%	75.12%	68.51%
User 5	87.12%	94.11%	85.52%	88.34%	71.81%	87.83%
User 6	76.78%	71.74%	79.24%	77.36%	69.19%	68.62%
User 7	76.76%	82.17%	70.18%	91.63%	67.18%	80.31%
User 8	85.12%	85.71%	90.97%	86.66%	73.47%	84.14%
User 9	84.90%	82.89%	92.92%	93.10%	85.33%	86.44%
User 10	70.35%	84.26%	84.24%	81.34%	81.99%	85.20%

TABLE 28. K-nearest neighbor (one-to-one model, oversampled).

	Arm	Hand	Pocket1	Pocket2	Waist	N
User 1	76.43%	78.09%	84.36%	80.76%	81.07%	78.46%
User 2	89.67%	68.94%	86.20%	92.37%	87.23%	86.86%
User 3	87.46%	76.90%	74.90%	77.36%	77.46%	78.92%
User 4	73.71%	61.51%	71.77%	72.24%	73.81%	72.14%
User 5	85.76%	91.48%	83.71%	89.38%	78.61%	88.09%
User 6	76.11%	72.87%	81.33%	79.16%	67.06%	74.85%
User 7	73.94%	81.58%	73.63%	88.36%	67.30%	80.08%
User 8	81.58%	83.58%	88.19%	87.09%	76.09%	81.37%
User 9	80.05%	83.39%	90.92%	90.92%	83.23%	88.02%
User 10	70.33%	86.77%	80.00%	85.03%	77.57%	83.93%

The result suggests that resampling methods are indispensable. The reason is apparent: on the one hand, in our daily life, the distribution of each activity tends to be unequal. For instance, walking activity appears more frequently than jumping and ascending/descending stairs, and sitting activity appears more frequently than jogging; on the other hand, activity recognition often applies multi-class algorithms, so the performance is governed by the majority, where data unbalance should be avoided.

VIII. DISCUSSION

Based on the findings from this study, we have acquired some useful messages, each of which may provide suggestions for future work. We also summarize some limitations of this work, in order to improve our work in the future.

A. SUGGESTIONS FROM THE RESULTS

Previous studies proposed various approaches and showed promising results in human activity recognition. After considering the success factors of their approach and experiment results in our work, here we present six aspects that may help to improve recognition performance.

First, an adequate amount of personalized data should be gathered. Because people differ in body size, gender, age and other physiological properties, people always show different movement pattern even for the same type of activity, and we can see everyone has his own *biological fingerprint*. In other words, only a user's own motion data can reflect his own movement accurately and completely. The fact that we got the worst performance in the *rest-to-one model* can be an evidence to prove our conclusion. This phenomenon brings us another inspiration: can we extract the *movement fingerprint* for authentication? This point will be explored in our future work.

Second, building *all-to-one model* is beneficial. It is worth noting that this point is not contrary to the former aspect. Personalized data is needed to promote the recognition accuracy, while generalized data can help to strengthen the stability to abnormal training data. Furthermore, training data from other users can help reduce the burden on data acquisition.

Third, our results suggest that applying more types of sensors can improve the accuracy. More sensors mean richer information, and more information means we can construct more fine-grained features. Besides of accelerometer and gyroscope, magnetometer is also a commonly used sensors embedded in smartphone, which has the potential to be an auxiliary sensor.

Fourth, we think data resampling method must be considered for smartphone-sensor-based activity recognition due to the different distributions of activities. By applying an appropriate oversampling technique, we can not only improve classifier's performance but also obtain more training data.

Fifth, the action detection and segmentation is based on the z-axis acceleration data. This method is easy to implement and efficient, but it may give false segmentation point when there are two different types of movement in the same time window. New action segmentation method needs to be investigated in the future.

Finally, we did not apply coordinate system transformation like [25] in our evaluation. As a result, even for the same position setting, the pattern may be not stable due to the slight variance of smartphone orientations, which might compromise the recognition accuracy. In our future work, we plan to employ coordinate system transformation method by including magnetic sensor data to improve the system's stability.

B. LIMITATIONS

Our work has tried to conduct a systematic performance evaluation for human activity recognition via smartphone motion sensors, but we have to admit that there are still some limitations in our work.

1) The dataset is relatively impoverished. As mentioned in Section III, the data collection procedure is not only tiring for us but also for these subjects. We can hardly acquire adequate data in a short time. Thus one aspect of our future work is to recruit more subjects into our work, so that we can analyze how the activity patterns differ in age, gender, status, and so on. Ideally, we hope we can establish a publicly available dataset for other researchers.

2) Physical condition will affect the movement. For instance, if someone climbs stairs to 10th floor, the speed tends to decrease as he climbs higher. However, to relieve subjects' discomfort in data collection procedure, we told them to have a rest if they felt tired. So although we requested the subjects to move at ease, the experiment was still slightly environment-controlled. In the following study, we are supposed to explore the variance of activities in different physical conditions.

3) Our research remains to be exploited for more practical applications. Inspired by some existing studies [31], [32], [33], [41], we can regard smartphone sensors as a widespread and mobile resources to perceive user's context, and using these information to detect emergency events by including an amount of users' data.

4) Our evaluation is performed on offline dataset. The performance of online system remains to be explored.

IX. CONCLUSION

Smartphone's sensory and computing ability has been improved significantly. Naturally, smartphone has become a perfect platform to perform human activity recognition. In this paper, we provide a brief review of related work in the human activity recognition field, and mainly focus on smartphone-sensor-based approaches. From our summary, we can see two main challenges: one is variety of smartphone position or orientation, and another is gross accuracy of embedded sensors. In view of the above-mentioned problems, we proposed a detailed HAR framework (see in Figure 7) that can perform human activity recognition with high accuracy via smartphone motion sensor. Time-, frequency- and wavelet-domain features were extracted to precisely characterize a subject's activity pattern. Then we employed two-sample K-S test to analyze motion sensor behavior, and we perform feature selection according to p-values. Our approach was evaluated on a dataset consisting of 27,681 samples from 10 subjects. To make a systematic evaluation, we implemented 4 multi-class classifiers: Random Forests (#tree $= 200$), Support Vector Machines (linear kernel and RBF kernel), and k Nearest Neighbor ($k = 3$). We compared the performance for different phone-placement settings, user spaces, and contributions of sensors.

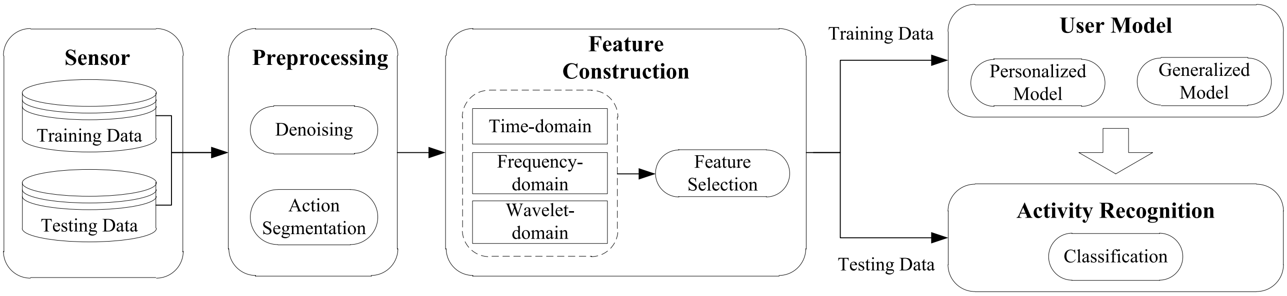

FIGURE 7. Activity recognition framework overview.

Furthermore, we investigated the efficiency of data resampling method.

Despite the fact that a large amount of work has been done in this area, from our results, we can see many issues remains open, such as movement segmentation method and online activity recognition technique.

APPENDIX

In this part, we provide the detailed results (in the form of F-score) in several tables. In these tables, character "N" represents "No position information" setting.

ACKNOWLEDGMENT

The authors are grateful to Ting Liu and Roy Maxion for his insightful comments and helpful advice, and to Z. Yang and H. Zhang for running the data collection. They would also like to thank several anonymous reviewers for their helpful comments.

REFERENCES

[1] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015.

[2] K. Park, H. Shin, and H. Cha, "Smartphone-based pedestrian tracking in indoor corridor environments," *Pers. Ubiquitous Computing*, vol. 17, no. 2, pp. 359–370, Feb. 2013.

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Ambient Assist. Living Home Care, 4th Int. Workshop, (IWAAL)*, Vitoria-Gasteiz, Spain, Dec. 2012, pp. 216–223.

[4] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, Oct. 2014.

[5] W.-H. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *Proc. 1st Int. Conf. Inf. Syst. Secur. Privacy (CISSP)*, Agers, France, 2015, pp. 270–280.

[6] L. Li, X. Zhao, and G. Xue, "Unobservable Re-authentication for smartphones," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Feb. 2013, pp. 1–16.

[7] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. Int. Conf. Pervasive Comput*. 2004, pp. 1–17.

[8] M. Ermes, J. PÄrkkÄ, J. MÄntyjÄrvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 26–27, Jan. 2008.

[9] Y. Xue and L. Jin, "Discrimination between upstairs and downstairs based on accelerometer," *IEICE Trans. Inf. Syst.*, vol. 94, no. 6, pp. 1173–1177, Jun. 2011.

[10] Z. He, Z. Liu, L. Jin, L. Zhen, and J. Huang, "Weightlessness feature—A novel feature for single tri-axial accelerometer based activity recognition," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, Tampa, FL, USA, Dec. 2008, pp. 1–4.

[11] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human activity recognition using multidimensional indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1091–1104, Aug. 2002.

[12] N. Robertson and I. Reid, "A general method for human activity recognition in video," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 232–248, Nov. 2006.

[13] S. Ke, H. Thuc, Y. Lee, J. Hwang, J. Yoo, and K. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, Jun. 2013.

[14] J. Farringdon, A. J. Moore, N. Tilbury, J. Church, and P. D. Biemond, "Wearable sensor badge and sensor jacket for context awareness," in *Proc. 3rd Int. Symp. Wearable Comput. (ISWC)*, San Francisco, CA, USA, Oct. 1999, pp. 107–113.

[15] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Proc. IEEE Int. Conf. Syst., Man, (SMC)*, vol. 2. Tucson, AZ, USA, Oct. 2001, pp. 747–752.

[16] S. H. Lee, H. D. Park, S. Y. Hong, K. J. Lee, and Y. H. Kim, "A study on the activity classification using a triaxial accelerometer," in *Proc. 25th IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, vol. 3. Cancún, NM, USA, Sep. 2003, pp. 2941–2943.

[17] M. Sekine, T. Tamura, M. Akay, T. Fujimoto, T. Togawa, and Y. Fukui, "Discrimination of walking patterns using wavelet-based fractal analysis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 3, pp. 188–196, Sep. 2002.

[18] X. Long, B. Yin, and R. M. Arats, "Single-accelerometer-based daily physical activity classification," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Minneapolis, MN, USA, Sep. 2009, pp. 6107–6110.

[19] N. Wang, E. Ambikairajah, N. H. Novel, and B. G. Celler, "Accelerometry based classification of walking patterns using time-frequency analysis," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Lyon, France, Aug. 2007, pp. 4899–4902.

[20] S. Wang, J. Yang, N. Chen, X. Chen, and Q. Zhang, "Human activity recognition with user-free accelerometers in the sensor networks," in *Proc. Int. Conf. Neural Netw. Brain, (ICNNB)*, vol. 2. Oct. 2005, pp. 1212–1217.

[21] A. M. Khan, Y. K. Lee, and T.-S. Kim, "Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Vancouver, BC, Canada, Aug. 2008, pp. 5172–5175.

[22] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring," *Comput. Human Behavior*, vol. 15, no. 5, pp. 571–583, Sep. 1999.

[23] Y. Lee and S. Cho, "Activity recognition using hierarchical hidden Markov models on a smartphone with 3D accelerometer," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst. (HAIS)*, Wroclaw, Poland, May 2011, pp. 460–467.

[24] T. Brezmes, J. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *Proc. Int. Workshop-Conf. Artif. Neural Netw. (IWANN)*, Salamanca, Spain, Jun. 2009, pp. 796–799.

[25] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," in *Proc. Int. Conf. Body Sensor Netw. (BSN)*, Dallas, TX, USA, May 2011, pp. 41–46.
[26] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Proc. 7th Inf. Conf. Ubiquitous Intell. Comput. (UIC)*, Xi'an, China, Oct. 2010, pp. 548–562.
[27] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newslett.*, vol. 12, no. 2, pp. 74–82, Dec. 2010.
[28] W. Wu, S. Dasgupat, E. E. Ramirez, C. Peterson, and G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *J. Med. Internet Res.*, vol. 14, no. 5, p. e130, Oct. 2012.
[29] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10146–10176, Jun. 2014.
[30] C.-C. Yang and Y.-L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.
[31] Z. Xu *et al.*, "Crowdsourcing based description of urban emergency events using social media big data," *IEEE Trans. Cloud Comput.*, to be published.
[32] Z. Xu *et al.*, "Building the search pattern of Web users using conceptual semantic space model," *Int. J. Web Grid Services*, vol. 12, no. 3, pp. 328–347, Jan. 2016.
[33] Z. Xu *et al.*, "Building knowledge base of urban emergency events based on crowdsourcing of social media," *Concurrency Comput. Pract. Exper.*, vol. 28, no. 15, pp. 4038–4052, Oct. 2016.
[34] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.
[36] O. Pujol, P. Radeva, and J. Vitria, "Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1007–1012, Jun. 2006.
[37] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*. New York, NY, USA: Springer, 2005, pp. 853–867.
[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[39] R. Liu, J. Zhou, M. Liu, and X. Hou, "A wearable acceleration sensor system for gait recognition," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Harbin, China, May 2007, pp. 2654–2659.
[40] M. O. Derawi, P. Bours, and K. Holien, "Improved cycle detection for accelerometer based gait authentication," in *Proc. 6th IEEE Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Darmstadt, Germany, Oct. 2010, pp. 312–317.
[41] Z. Xu *et al.*, "From latency, through outbreak, to decline: Detecting different states of emergency events using Web resources," *IEEE Trans. Big Data*, to be published.

YUFEI CHEN is currently pursuing the degree with the State Key Laboratory of Mechanical Manufacturing Systems Engineering Laboratory, Systems Engineering Institute, Xi'an Jiaotong University. His research interests include computer security and insider/intrusion detection.

CHAO SHEN (S'09–M'14) is currently an Associate Professor with the School of Electronic and Information Engineering, Xi'an Jiaotong University of China. His research interests include insider detection and behavioral biometric.