A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities

Wen Qi, *Student Member, IEEE*, Hang Su, *Member, IEEE*, and Andrea Aliverti

***Abstract*—Human activity recognition (HAR) using smartphones provides significant healthcare guidance for telemedicine and long-term treatment. Machine learning and deep learning (DL) techniques are widely utilized for the scientific study of the statistical models of human behaviors. However, the performance of existing HAR platforms is limited by complex physical activity. In this article, we proposed an adaptive recognition and real-time monitoring system for human activities (Ada-HAR), which is expected to identify more human motions in dynamic situations. The Ada-HAR framework introduces an unsupervised online learning algorithm that is independent of the number of class constraints. Furthermore, the adopted hierarchical clustering and classification algorithms label and classify 12 activities (five dynamics, six statics, and a series of transitions) autonomously. Finally, practical experiments have been performed to validate the effectiveness and robustness of the proposed algorithms. Compared with the methods mentioned in the literature, the results show that the DL-based classifier obtains a higher recognition rate (95.15%, waist, and 92.20%, pocket). The decision-tree-based classifier is the fastest method for modal evolution. Finally, the Ada-HAR system can monitor human activity in real time, regardless of the direction of the smartphone.**

***Index Terms*—Data compression, deep learning (DL), hierarchical classification (HC), human activity recognition (HAR).**

I. Introduction

RECOGNIZING human activities using inertial sensors in a smartphone has attracted increasing research interests during the past decades in various domains, ranging from home healthcare to sports monitoring, rehabilitation, personalized medicine, and mental disorders [1]. Recently, with the development of the Internet of Things, machine learning (ML), and deep learning (DL) techniques, human activity recognition (HAR) can be achieved by transferring data in the body area networks and wireless Ethernet, allowing the assessment of the human physical and physiological status [2]. However, many studies develop appropriate tasks for a given HAR system by resorting to extensive heuristic knowledge [3]. They are applicable in a laboratory or well-controlled situation via body-fixed mobile devices [4]. The variability of the disturbances of the mobile devices affects the recognition rate of the HAR system, such as movement artifacts, baseline noise, the happening of new activity, and the differences among users. For example, identifying the same walking status of the elderly and youths is challenging. The carefully engineered features are challenging to model intricate activity details and are time-consuming. Hence, it is interesting to introduce an adaptive classifier for identifying activities in a dynamic situation [5].

In this article, we propose a smartphone-based adaptive HAR real-time monitoring system (Ada-HAR) to identify 12 human activities. These include five dynamic exercises (i.e., walking, jogging, jumping, go upstairs, and go downstairs), seven static postures (i.e., standing, sitting, lying to the left and right sides, lying prone, and lying supine), and a series of transitions (e.g., from sitting to standing). The system aims to label the raw signals automatically by using the hierarchical k-medoids clustering (Hk-mC) algorithm [6] and builds a hierarchical classification (HC) [7] for HAR. Finally, the developed Ada-HAR system is validated with a group of subjects for monitoring human activities in both remote and local LANs. The main contributions of the proposed Ada-HAR system are outlined as follows.

1) It can update the previous classifier in a dynamic situation.
2) The data compression method achieves fast computation.
3) The proposed signal processing approaches avoid the disturbances from the changes in position and orientation.

The article is structured as follows. Section II introduces the state-of-the-art HAR monitoring systems or algorithms. Section III describes the general protocol and problem of the Ada-HAR system by functions and notations. Section IV introduces the structure of Ada-HAR system by explaining the details and algorithms of each module. Section V shows the results of three experiments to evaluate the performance of Ada-HAR system. Section VI discusses the achievements and delineates avenues for further work on this topic. Section VI concludes this article.

II. Related Work

The pioneering works of recognizing human activity using smartphones explored different sensing technologies and feature extraction methods for improving the identification ability. Wang *et al.* [8] proposed a hybrid filter and wrapper (HFW) method to extract more useful features from accelerometer and gyroscope. The HFW approach was proved to obtain higher accuracy than principal component analysis (PCA) [9], fast

Manuscript received May 24, 2019; revised September 2, 2019 and January 1, 2020; accepted February 23, 2020. Date of publication April 24, 2020; date of current version September 15, 2020. This article was recommended by Associate Editor Bin Guo. *(Corresponding author: Wen Qi.)*

The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano, Italy (e-mail: wen.qi@polimi.it; hang.su@polimi.it; andrea.aliverti@polimi.it).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2020.2984181

correlation-based filter [10], and sequential forward selection (wrapper) [11] methods. By comparing the classification rate of identifying five daily activities carrying a smartphone on five body positions, Khan *et al.* [12] proved that Kernel discriminant analysis is better than linear discriminant analysis and signal magnitude area. Shoaib *et al.* [13] demonstrated putting three types of sensors (i.e., accelerometer, gyroscope, and linear accelerometer) on the wrist and pocket to identify less-repetitive activities, such as smoking, eating, and drinking coffee. For solving the problem of online time series segmentation, Ignatov *et al.* [14] constructed the phase trajectory matrix and applied the PCA technique to extract features of the first period. Sousa *et al.* [15] investigated the classification capability with different feature sets, such as time-domain (mathematical and statistical parameters), frequency-domain (wavelet transform), and discrete-domain (symbolic representations). Although the above works achieved significant progress for HAR, the limitations still existed. Most of them ignored the affection of shift changes in position and orientation of the smartphone due to motion artifacts.

During the past decades, many researchers have explored various ML and DL algorithms to identify more complex activities with higher recognition rates. Kose *et al.* [1] proved that k-nearest neighbor (k-NN) could obtain a higher classification accuracy than Naïve Bayes (NB) by recognizing four kinds of activities, i.e., walking, running, sitting, and standing. Lee and Cho [16] designed a mixture-of-experts model for dealing with uncertain and incomplete data. Reyes-Ortiz *et al.* [17] proposed a transition-aware HAR system based on the support vector machine (SVM) algorithm to classify a broad spectrum (up to 33 types) of activity with smartphone in real-time. Lee and Cho [18] presented a hierarchical hidden Markov model (HHMM) method to classify motions (standing, walking, go stairs, and running) and daily activities (shopping, taking a bus, and moving arms). A two-stage continuous hidden Markov model (HMM) algorithm was proposed to reduce the number of feature subsets because it decomposed the complex activities into several simpler ones [19]. Reiss *et al.* [20] found that the ConfAdaBoost.M1 ensemble algorithm could obtain a high recognition accuracy, but it was necessary to extract 561 features. Noor *et al.* [21] introduced an adaptive sliding window approach to recognize physical activity by computing the probability of signals with an adjustable window length. All of the above contributions were limited by the length of the detected signal segments. Some of them focused on a short segment, which cannot express a whole activity. Some of them detected on the long signal segment, which affected misclassification due to a segment including many activities. Hence, it is essential to find a suitable data length for classification. However, the performance of ML-based solutions was strongly dependent on the extracted features from the used sensors. Different features were utilized for achieving several kinds of HAR problems [22]. An efficient group-based context-aware classification method has been proposed to improve the classification performance by exploiting hierarchical group-based scheme and context awareness rather than the intensive computation [23].

Recently, the advancement of HAR with inertial measurement unit (IMU) sensor broadly adopts DL algorithms. Ronao and Cho [24] designed a three-(and four-) layer convolutional neural networks (CNN) with the short-time fast Fourier transform method for classifying six activities with a high classification accuracy by comparing with the traditional ML algorithms, i.e., NB, ANN, and j48 decision tree (DT). Hammerla *et al.* [25] proved that bidirectional long short-term memory (Bi-LSTM) obtained more accurate results than deep NN, CNN, and other recurrent neural network (RNN) by testing on three public datasets. Although the DL-based approaches were widely known to extract features automatically, they were affected by the selected sensors. Wang *et al.* [8] compared the performance obtained using different inertial sensors in a smartphone. Using both accelerometer and gyroscope was demonstrated to yield accuracy enhancement. In our previous work, we compared the recognition rate for identifying 12 activities by using different combined sensors, such as accelerometer, gyroscope, magnetometer, and orientation. By combining accelerometer, gyroscope, and direction, the designed deep CNN model could obtain high accuracy [26], [27]. The results proved that adopting transferred signals and more sensors can perform the results. Moreover, the DL-based methods were time-consuming for classifier evolution in a dynamic situation [28].

III. Problem Statement

The following notations describe the data stream of the Ada-HAR system through training, testing, and updating procedures, as shown in Fig. 1. The embedded IMU sensors in a smartphone can provide nine-dimensional (9-D) raw signals S, including acceleration, magnetism, and angular velocity [29]. S can be divided into M segments, namely $s_p, p = 1, 2, \ldots, M$, by the sliding window method [30]. The segments can be labeled in a supervised manner as $\{s_p, y_p\}, p = 1, 2, \ldots, M$. A data compression model is adopted to remove the similar segments from $\{s_p, y_p\}$ with the same labels. The compressed datasets $\{s_i, y_i\}, i = 1, 2, \ldots, N$ (usually, $N \leq M$) can be processed by the signal preprocessing model for information enhancement as $\{s_i^*, y_i\}$. The proposed Ada-HAR system can build the classifier $f(X, \theta)$ based on random ML or DL algorithms. The input X can be either the extracted features Fh_{s^*} or the processed signals s^*. Meanwhile, the dataset $\{s_i^*, y_i\}$ will be reconstructed as a basic dataset $s_{qk}^*, k = 1, 2, \ldots, n_c; q \in \mathbb{R}^+$ (where n_c is the number of classes and qk is the label), for comparing with new activity.

In the testing procedure, the classifier predicts a current activity $\hat{y}_t$ based on the extracted feature $Fh_{s_t^*}$ (ML-based) or the processed signal s_t^* (DL-based). The classification error ε can be calculated through a supervised learning strategy by comparing $\hat{y}_t$ with real label y_t as follows:

$$\varepsilon(s, y, X) = \frac{1}{N} \sum_{t=1}^{N} [y_t \neq f(X, \theta)] \tag{1}$$

where

$$[y_t \neq f(X, \theta)] = \begin{cases} 1, & y_t \neq f(X, \theta) \\ 0, & y_t = f(X, \theta) \end{cases}. \tag{2}$$

Let Θ be the overall set of classifier parameters. The Ada-HAR system aims to find the optimal parameter set θ via the following

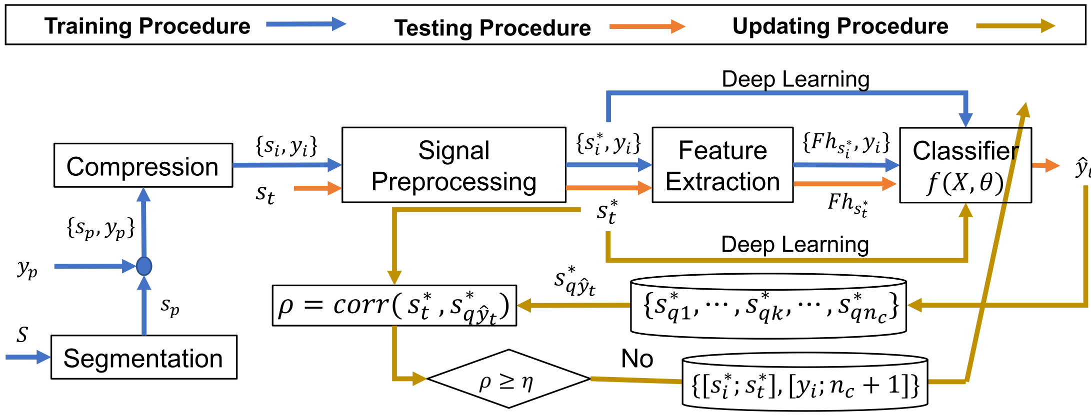

Fig. 1. Data stream of Ada-HAR system. It includes training (blue line), testing (orange line), and updating (light tan line) procedures.

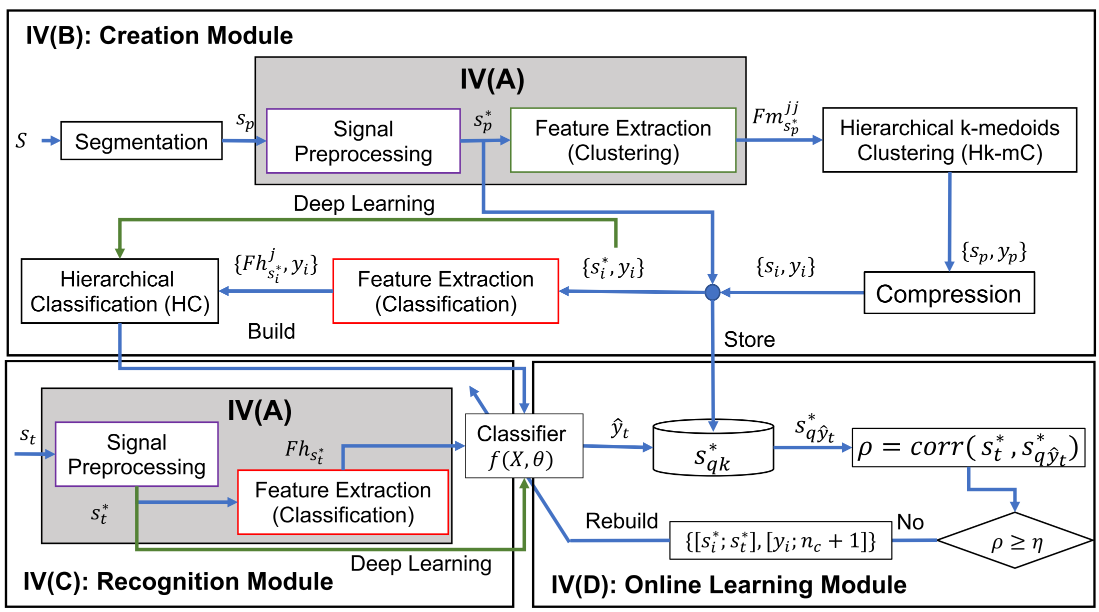

Fig. 2. Architecture of the adaptive real-time human activity recognition monitoring system (Ada-HAR). It includes a creation module IV(B) to label activities and builds the classifier, a recognition module IV(C) to predict human activity in an online manner, and an online learning module IV(D) to update the classifier in an unsupervised manner. Boxes with the same color adopt the same algorithms, such as the feature extraction (classification) model in the creation and recognition modules. The shaded part is Section IV(A): Signal preprocessing and feature extraction.

equation:

$$\arg\min_{\theta \in \Theta} \varepsilon(s, y, X). \tag{3}$$

In the updating procedure, the new processed input s_t^* is compared with the training dataset $s^*_{q_{\hat{y}_t}}$ by calculating the correlation coefficient $\rho = \mathrm{corr}(s_t^*, s^*_{q_{\hat{y}_t}})$. If $\rho < \eta$ (we set $\eta = 0.8$), the Ada-HAR system will retrain the classifier on the updated training dataset $\{[s_i^*; s_t^*], [y_i; n_c + 1]\}$.

IV. Methodology

Fig. 2 shows the proposed Ada-HAR system with creation, recognition, and online learning modules. In the creation module, the Hk-mC [6] is adopted to label the activities autonomously. The HC [7] algorithms are used to establish the classifier for recognizing the 12 original activities (described in Section I). In the recognition module, the obtained HC classifier is implemented for HAR in real time by carrying a smartphone on the waist or putting it in the left pant pocket. In the online learning module, a new activity that is not included in the 12 original activities will be identified in an unsupervised learning manner. Meanwhile, the old classifier is updated. The signal preprocessing and feature extraction models are shared in both creation and recognition modules.

A. Signal Preprocessing and Feature Extraction

The collected signals typically exhibit various turbulence (e.g., magnetic fields and movement artifact), which affects the recognition ability of the classifier. The derived sensors signals (e.g., gravity and linear acceleration) that are transferred by the

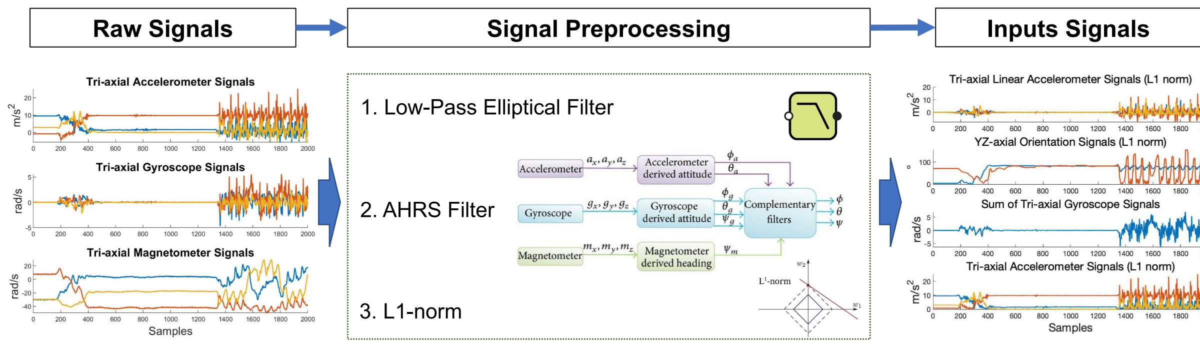

Fig. 3. Schematic diagram of signal preprocessing model. The 9-D raw signals [S_α (accelerometer), S_β (gyroscope), and S_γ (magnetometer)] are processed by the L1-norm (Manhattan distance) transfer, third-order zero phases low-pass elliptical filter (LPEF), attitude and heading reference system algorithm (AHRS filter), and sum of gyroscope signals algorithms to provide the inputs, namely, the linear L1 norm acceleration $|S_\alpha|_l$, *YZ*-axis L1 norm orientation $|S_{\ominus_{YZ}}|$, sum of gyroscope $\sum S_\beta$, and L1-norm accelerometer signals $|S_\alpha|$.

original IMU sensors, i.e., S_α (accelerometer), S_β (gyroscope), and S_γ (magnetometer), can solve the problems.

First, the linear L1-norm accelerometer $|S_\alpha|$ [by (4)] can fix the three axes of acceleration for overcoming the influence of the changes in direction and position

$$||S_\alpha|| = \sum_{i=1}^{L_d} |s_{\alpha_i}| \tag{4}$$

where s_α is the subsegment of S_α with L_d length. Then, a third-order zero phase low-pass elliptical filter (LPEF) [31] is implemented to decompose $|S_\alpha|$ into the gravity and the linear acceleration ($|S_\alpha|_l$) vectors to remove the high-order noises.

Second, the attitude and heading reference system algorithm (AHRS filter) [32] calculates the orientation axes $S_\ominus$ and determine the smartphone's reference system to increase the identification ability on static activities. Instead of tracking the orientation directly, the indirect Kalman filter models the error process x with a recursive update as follows:

$$x_t = \begin{bmatrix} S_{\ominus_t} \\ S_{\alpha_t} \\ S_{\beta_t} \\ S_{\gamma_t} \end{bmatrix} = F_t \begin{bmatrix} S_{\ominus_{t-1}} \\ S_{\alpha_{t-1}} \\ S_{\beta_{t-1}} \\ S_{\gamma_{t-1}} \end{bmatrix} + \omega_t \tag{5}$$

where the 12-by-1 vector x_t is the output at time t, consisting of the original three sensors and the orientation composition $S_{\ominus_t}$. ω_t is 12-by-1 additive noise vector and F_t is the state transition model. Third, the sum of angular velocity, namely $\sum S_\beta = S_{\beta_x} + S_{\beta_y} + S_{\beta_z}$, can extract more dynamical information.

For accuracy enhancement, we select parts of the signals as the outputs of signal preprocessing model. They are the tri-axial linear L1 norm acceleration ($|S_\alpha|_l$), *YZ*-axis L1 norm orientation ($|S_{\ominus_{YZ}}|$), the sum of gyroscope signals ($\sum S_\beta$), and the L1-norm accelerometer ($|S_\alpha|$), namely $S^* = [|S_\alpha|_l; |S_{\ominus_{YZ}}|; |S_{\ominus_{YZ}}|; \sum S_\beta |S_\alpha|]$ (shown in Fig. 3).

In addition, it is better to extract effective features (Table I) for building an ML-based classifier and divide them into a dynamical set $\mathbb{D}$, a static set $\mathbb{S}$, and a single feature $\mathbb{D}/\mathbb{S}$: $\sigma(\sum ||S_\alpha|_l|^2))$ for establishing an HC classifier. $|S_\alpha|_l$ and $\sum S_\beta$ are used to provide dynamical features, while $|S_{\ominus_{YZ}}|$ and $|S_\alpha|$ are selected to compute the static features. The meanings of the symbols are as follows: Standard deviation σ, average (e.g., $\bar{\sum} S_\beta$), maximum $\max$, minimum $\min$, difference (recurrence relation) $\bigtriangledown$, and the whole numbers ($\mathbb{N}$).

TABLE I
THE EXTRACTED FEATURES IN TIME DOMAIN

Label	Features
$\mathbb{D}/\mathbb{S}$	$\sigma(\sum \Vert S_\alpha\vert_l\vert^2))$
$\mathbb{D}$	$\sigma, \max, \min([\vert S_\alpha\vert_l; \sum S_\beta; \Vert S_\alpha\vert_l\vert; \vert \sum S_\beta\vert; \vert S_\alpha\vert_l^2; \sum S_\beta^2]),$ $\vert \bar{S}_\alpha\vert_l, \bar{\sum} S_\beta, \Vert \bar{S}_\alpha\vert_l\vert, \vert \bar{\sum} S_\beta\vert, \vert \bar{S}_\alpha\vert_l^2, \bar{\sum} S_\beta^2$ $\bigtriangledown([\bar{S}_{\alpha_l}; \sum S_\beta])]), \mathbb{N}(\bigtriangledown([\bar{S}_{\alpha_l}; \sum S_\beta]) > 1)$
$\mathbb{S}$	$\max, \min([\vert S_{\ominus_{YZ}}\vert; \vert S_\alpha\vert]),$ $\vert \bar{S}_{\ominus_Y}\vert, \vert \bar{S}_{\ominus_Z}\vert, \vert(\vert \bar{S}_{\ominus_Y}\vert - \vert \bar{S}_{\ominus_Z}\vert)\vert$

B. Creation Module

The creation module includes two steps: One aims to achieve automatic labeling using the Hk-mC algorithm, and the other one is to build the HC classifier for HAR. The Hk-mC algorithm 1 is implemented in an unsupervised manner with the fixed structure. It labels the extracted features set $Fm_{s_p^*}^{jj}, jj = 1, 2, \ldots, Ly_m$ computed on the processed segments $s_p^*, p = 1, 2, \ldots, M$ (lines 2 and 3). For each layer, it calculates a partition of a dissimilarity object consisting of $kk \le n_c$ medoids. Given a set of M data objects $S = s_p$, the objective is to partition the dataset into n_c classes, with each cluster having one representative object known as a medoid s_{jj}^m [33]. The objective function of the k-medoids clustering is

$$\text{Cost}_{kk} = \sum_{jj=1}^{Ly_m} \sum_{s_p \in M} D_{kk}(s_p, s_{jj}^m) \tag{6}$$

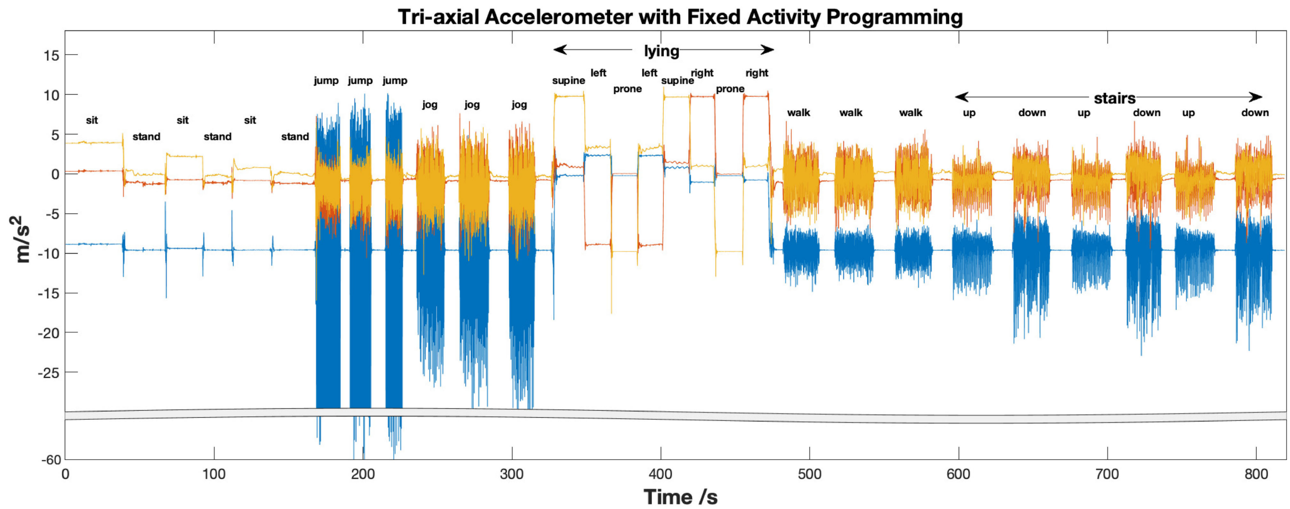

Fig. 4. Labeled tri-axis acceleration with fixed activity programming.

TABLE II
THE LAYERS AND FEATURES FOR CLUSTERING

Activity	Layers	Features
$\mathbb{D}/\mathbb{S}$	Fm^1	$\sigma(\sum \|\|S_\alpha\|_l\|^2))$
Jump, Jog, Others	Fm^2	$\sigma, \max, \min(\|S_\alpha\|_l)$
Downstairs, Upstairs, Others	Fm^3	$\sigma(\|S_{\ominus YZ}\|), \|\bar{S}_{\ominus Y}\|, \|\bar{S}_{\ominus Z}\|$
Transition, Walk	Fm^4	$\sigma, \max(\nabla(\|S_{\ominus YZ}\|))$
Stand, Sit, Others	Fm^5	$\|\bar{S}_{\ominus Y}\|, \|\bar{S}_{\ominus Z}\|$
Lying on four positions	Fm^6	$\max, \min(\|S_\alpha\|), \|\bar{S}_{\ominus Y}\|, \|\bar{S}_{\ominus Z}\|$

Algorithm 1: Hierarchical *k*-medoids Clustering (Hk-mC).

Input:
the raw IMU signals S, the number of layers Ly;
Output: labeled signal segments $\{s_p, y_p\}$;
1: divide S into M segments $\{s_p\}$;
2: acquire the processed signals s_p^* by the algorithms in section IV-A;
3: extract features $Fm_{s_p^*}^{jj}$ based on Table II;
4: label the features and corresponding segments as $\{s_p, y_p\}$;

where D_{kk} is the dissimilarity between data object s_p and medoid s_{jj}^m associated with cluster jj. In this article, D_{kk} is the city block distance $D_{kk}(s_p, s_{jj}^m) = |s_p - s_{jj}^m|$. Then, the algorithm clusters the features into kk classes and labels the segments as $\{s_p, y_p\}$ (line 4).

For clustering accuracy enhancement, the subjects were asked to do the 12 activities with a fixed order (shown in Fig. 4). In addition, the number of clustering layers and order were fixed and are reported in Table II. The mislabeled activities using the Hk-mC algorithm will be removed or corrected manually.

However, similar segments increase the computing burden for building and updating a classifier. To solve this problem for fast computation, a correlation-based data compression algorithm is designed to remove partially similar segments based on the similarity measurement theory [34]. The data compression algorithm aims to compare similarity by computing the covariance coefficient ρ (7) between every two matrices s_a and s_b in the reconstructed dataset $s_{pk}, k = 1, 2, \ldots, n_c$ with the same label, where $a, b \in \mathbb{R}^+$ and $a \neq b$. For saving the compressing time, we adopt the seeker optimization algorithm [35] to control the speed by setting a trust degree η. If $\rho \leq \eta$, one of the two segments (s_a and s_b) will be deleted. In this article, $\eta = 0.98$.

$$\begin{aligned}\rho &= \mathrm{corr}(s_a, s_b)\\ &= \frac{\sum_{ii}(s_{a_{ii}} - \bar{s_a})(s_{b_{ii}} - \bar{s_b})}{\sqrt{\sum(s_a - \bar{s_a})^2}\sqrt{\sum(s_b - \bar{s}_{bii})^2}}\\ &= \frac{\langle s_a - \bar{s_a}, s_b - \bar{s_b}\rangle}{||s_a - \bar{s_a}||\,||s_b - \bar{s_b}||}.\end{aligned} \tag{7}$$

Finally, the compressed training dataset $\{s_i, y_i\}, i = 1, 2, \ldots, N$ is used to train the classifier $f_j(X, \theta^j), j = 1, 2, 3$ based on random ML or DL algorithms. The first binary classifier $f_1(X, \theta^1)$ is to divide static and dynamical activities by the single feature $\sigma(\sum ||S_\alpha|_l|^2))$ (line 1). The second six-class classifier $f_2(X, \theta^2)$ separates the dynamical activities based on the feature set $\mathbb{D}$ reported in Table I (ML-based) or the labeled segments $\{s_i^*, y_i\}$ (DL-based). Similarly, the third classifier $f_3(X, \theta^3)$ divides the seven static activities by the feature set $\mathbb{S}$ (ML-based) or the labeled segments $\{s_i^*, y_i\}$ (DL-based) in line 3.

Five ML algorithms (i.e., *k*-NN, DT, NB, ANN, and SVM) and a DL method (LSTM) are chosen. As a nonparametric approach, the *k*-NN algorithm can search a group of *k* samples that are nearest to unknown samples based on distance functions. In this article, we adopt the city block distance. The computed class attributes of the *k*-NN in these *k* samples determined the labels. As a result, the parameter *k* was defined utilizing a bootstrap procedure. However, the *i*-NN method needs to handle missing

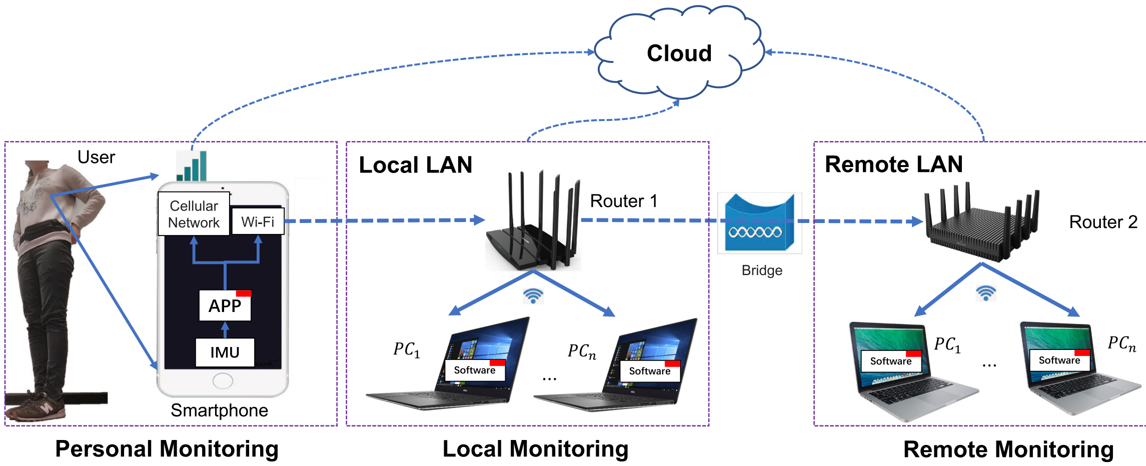

Fig. 5. Schematic diagram of the Ada-HAR communication system. In the "personal monitoring" unit, the subject's activity can be identified by carrying the smartphone where it has installed software with the proposed algorithms. In the "local monitoring" unit, people can monitor the users' activity by analyzing the received data from the Wi-Fi network. In the "remote monitoring" unit, the predicted activities can be monitored by receiving the signals in a multipoint wireless bridge connection network.

Algorithm 2: Hierarchical Classification (HC).

Input: the labeled signal segments (s_i^*, y_i) or features set $\{Fh^j_{s_i^*}, y_i\}$;

Output: the three layers HC classifier $f_j(X, \theta^j), j = 1, 2, 3$;

build a binary classifier $f_1(X, \theta^1)$ to classify dynamic and static activities by the feature $\sigma(\sum ||S_\alpha|_l|^2))$;

train a six-class classifier $f_2(X, \theta^2)$ to identify dynamical activities;

establish a seven-class classifier $f_3(X, \theta^3)$ to divide the static activities;

data. It is sensitive to class outliers and lots of irrelevant attributes [36]. Different from the *k*-NN method, the DT approach needs to set two parameters, namely the number of trees and the number of features in each split. A large number of trees can provide a robust model for predicting real-world results [37]. However, this operation should optimize more parameters such that it will be time-consuming. The NB approach assumes that all variables are mutually correlated and contribute towards classification. When the class variable is given, the NB classifier considers that the presence (or absence) of a particular attribute of a class is unrelated to the presence (or absence) of any other features. However, the strong assumption on the shape of data distribution limits the performance of the built classifier [38]. The ANN method is a widely used method for classification [39]. It combines several processing layers using simple elements operating in parallel to build the connection between inputs and outputs. However, the problems of overfitting and underfitting bring too much uncertainty learning in a dynamic situation. The SVM algorithm is to distinctly classify the signal segments s_i^* by finding a hyperplane in an *N*-dimensional (*N*-D) space. There are two primary parameters of the SVM approach, namely the optimum parameters of cost and the kernel width parameter.

Algorithm 3: The recognition algorithm.

Input: the HC classifiers $f(X, \theta)$, current input s_t, sample frequency Fs, detection length L_d, and overlay L_o;

Output:

predicted activities $\hat{y}_t$;

1: **while** $(t \cdot Fs)/(L_d + L_o) = 1$,
2: identify dynamic or static activity by $\hat{y}_t = f^1(X, \theta^1)$ based on feature $\sigma(\sum ||S_\alpha|_l|^2))$;
3: **if** $\hat{y}_t \in \Lambda_D$, identify a dynamical activity on features $\mathbb{D}$ or segment s_t^*;
4: **else** $\hat{y}_t \in \Lambda_S$, identify a static activity on features $\mathbb{S}$ or segment s_t^*;
5: **end if**
6: **end while**

For accuracy enhancement, the SVM model commonly utilizes the radial basis function kernel and linear kernel commonly. As a standard RNN method, the LSTM algorithm learns the long-term dependencies based on the three gates, namely an input gate, a forget gate to allow the LSTM unit to unlearn the previous memory, and an output gate to decide the quantity of memory transferring to the next hidden layers. However, it will cost ample time to optimize the parameters of the LSTM network and predict a result.

C. Recognition Module

Fig. 5 shows the details of the communication network for monitoring daily human activities in real time with a local LAN and a remote LAN. Algorithm 3 describes the procedure for identifying human activities. When a new input s_t is divided as a dynamical or static activity by the first classifier $\hat{y}_t = f^1(X, \theta^1)$ (line 1), it will be further identified as a dynamical or a static activity by the second or third classifier (lines 3 and 4).

Algorithm 4: The online learning algorithm.

Input: the old HC classifiers $f(X,\theta)$ and new predicted input $\{s_t^*, \hat{y}_t\}$;
Output: the updated HC classifiers $\hat{f}(X,\hat{\theta})$;
1: compute covariance coefficients $\rho_r = \text{corr}(s_t^*, s_{q\hat{y}_t}^*)$;
2: **if** $\rho_r \geq \eta$
3: update training dataset $\{[s_i^*; s_t^*], [y_i; n_c + 1]\}$;
4: **if** $\hat{y}_t \in \Lambda_D$
5: retrain the second classifier $\hat{f}^2(X,\hat{\theta})$ on $\{F_{[s_i^*;s_t^*]}, [y_i; n_c+1]\}$ (ML-based) or $\{[s_i^*; s_t^*], [y_i; n_c + 1]\}$ (DL-based);
6: **else** $\hat{y}_t \in \Lambda_S$
7: retrain the third classifier $\hat{f}^3(X,\hat{\theta})$ on $\{F_{[s_i^*;s_t^*]}, [y_i; n_c+1]\}$ (ML-based) or $\{[s_i^*; s_t^*], [y_i; n_c + 1]\}$ (DL-based);
8: **end if**
9: **else** continue **end if**

D. Online Learning Module

The online learning module aims to update the previous HC classifier when a new activity is found. An unsupervised learning approach (Algorithm 4) is proposed to measure the similarity degree between the new inputs s_t^* and saved training dataset s_{qk}^*. When the predicted activity $\hat{y}_t$ is available, the similarity measurement mechanism will compute the correlation coefficients ρ sequentially between s_t^* and the saved training set $s_{q\hat{y}_t}^*$ with the same label, namely, $\rho_r = \text{corr}(s_t^*, s_{q\hat{y}_t}^*), r = 1, 2, \ldots, q\hat{y}_t$ (line 1). If $\rho_t < \eta$ is found, the classifier will be rebuilt on the updated training dataset $\{[s_i^*; s_t^*], [y_i; n_c + 1]\}$, where the new training dataset includes the new input s_t^* with a new digital label $n_c + 1$ (line 3). In this article, $\eta = 0.8$.

V. Experiments and Results

Three experiments were designed to validate the claims about the Ada-HAR system in Section I. The first experiment of the HC algorithm evaluation aimed to compare the performance among ML-based and DL-based HC classifiers. The second experiment was to validate the evolution ability of the Ada-HAR system. The demonstration revealed the identification capability of Ada-HAR system in real time. The experiments have implemented these methods in MATLAB 2018b with the hardware platform of Intel(R) i7 Core, 2.80-GHz CPU, and 16.0-GB RAM. A 64-bit iPhone 6 s (2-core CPU) was utilized for collecting data with a 50-Hz sampling frequency.

A. HC Algorithm Evaluation

Twenty-five subjects (13 females and 12 males, age range 18–40 years old) were asked to perform the 12 original activities described in Section I in a fixed order (shown in Fig. 4). They carried the smartphone on the waist or put it in their left pant pocket, respectively. Finally, 50 datasets (25 datasets for each position) were collected for further analysis.

We adopt the leave-one-out cross-validation strategy to validate the performance of HC classifiers [40]. The length of the detection window is 150 samples (3 s), and the overlap is 1 s (50 estimates). Table III reports a comparison of the average and standard deviation in terms of the accuracy, training time, and average test time among ML-based (i.e., *k*-NN, DT, NB, ANN, and SVM) and DL-based (LSTM) HC classifiers. Meanwhile, all of the HC classifiers are evaluated on uncompressed and no-compressed datasets. Although the LSTM-based HC classifier obtains the highest accuracy, it is time-consuming for building the model. Due to the SVM-based HC algorithm adopting the linear kernel function, it takes more time to search the optimal solution. In the same case, the *k*-NN-based approach is the fastest method to establish the HC classifier, while the DT-based classifier is the fastest algorithm to predict the activity. Moreover, the testing accuracy displays that putting the smartphone in the pocket acquires more noises than carrying the smartphone on the waist. The results for the average testing time and training time computed on the compressed dataset are significantly less than those for the uncompressed dataset. Hence, compressing the training datasets saves more time for building a new classifier. Even if the accuracy obtained on the uncompressed dataset is greater than that on the compressed dataset, it is time-consuming to build the classifier and predict an activity. Moreover, the trust degree η can be adjusted to increase the accuracy using (7).

B. Online Learning Examination

This experiment aims to prove the updating ability of Ada-HAR system for recognizing a new activity. Two new activities (squats and twisting hips) were collected from one subject carrying the smartphone on the waist or put in the left pocket, respectively. To confirm that the proposed signal processing algorithms could overcome the effect of changing the direction of mobile phone, we combined the collected data of two directions with the same testing protocol (shown in each top graph of Fig. 6). The DL-based HC algorithm was not adopted because the training procedure is costly. The experiments adopted an online batch learning mechanism. The SVM method used the Gaussian kernel. The built ML-based classifiers for identifying the 12 original activities were used to predict the new activity on the four datasets. The following four experiments were performed to evaluate the adaptive performance of both compressed and uncompressed datasets.

Fig. 6(a) displays a comparison results of the online accuracy and computational time for predicting squats by carrying the mobile phone on the waist. The top graph is the collected 3-axis acceleration with two directions. The middle graph is the online classification accuracy computed by (1) and (2) at time (epoch) t. Both *k*-NN and SVM methods obtain a higher accuracy than the other methods. The bottom graph is the online predicting time. The amplitudes of *k*-NN, DT, and SVM are 0.05 s, while the other approaches require more than 1 s to predict activity. Hence, the *k*-NN-based HC classifier is the best method for recognizing the new activity. Fig. 6(b) portrays the results of predicting squats by putting the mobile phone in the left pant pocket. The HC classifiers based on *k*-NN, SVM, and DT obtain a higher accuracy for identifying squats, and *k*-NN spends less time to retrain the classifier. Fig. 6(c) and (d) shows the validation

TABLE III
THE COMPARATIVE RESULTS AMONG HC CLASSIFIER BASED ON K-NN, DT, NB, ANN, SVM, AND LSTM

	HC classifier with	Test Accuracy (%)		Train Time (s)		Average Test Time (s)	
		Waist	Pocket	Waist	Pocket	Waist	Pocket
Compression	k-NN	87.02 ± 3.93	83.09 ± 4.49	$\mathbf{0.0352} \pm 0.0020$	$\mathbf{0.0491} \pm 0.0025$	0.0035 ± 0.0001	0.0042 ± 0.0002
	DT	86.88 ± 3.81	82.02 ± 3.81	0.1121 ± 0.2427	0.2387 ± 0.0327	$\mathbf{0.0013} \pm 0.0001$	$\mathbf{0.0015} \pm 0.0001$
	NB	81.13 ± 3.69	79.89 ± 5.08	4.0146 ± 3.8815	4.6738 ± 2.8099	0.2184 ± 0.1354	0.2260 ± 0.1395
	ANN	86.19 ± 3.71	85.74 ± 3.64	6.8158 ± 1.3242	28.437 ± 4.9586	0.0131 ± 0.0010	0.0112 ± 0.0011
	SVM	83.55 ± 8.67	80.87 ± 5.09	1.5284 ± 0.4350	722.91 ± 49.37	0.0052 ± 0.0004	0.0062 ± 0.0007
	LSTM	$\mathbf{92.93} \pm 3.32$	$\mathbf{88.37} \pm 4.87$	1056 ± 80.84	1512 ± 419.85	0.0128 ± 0.0012	0.0131 ± 0.0014
No Compression	k-NN	92.35 ± 3.81	83.91 ± 4.54	$\mathbf{0.1048} \pm 0.0097$	$\mathbf{0.1060} \pm 0.0093$	0.0098 ± 0.0006	0.0093 ± 0.0006
	DT	91.67 ± 3.95	83.49 ± 4.42	0.8071 ± 0.0904	1.1598 ± 0.1875	$\mathbf{0.0017} \pm 0.0002$	$\mathbf{0.0018} \pm 0.0002$
	NB	89.60 ± 2.67	81.23 ± 5.18	6.5750 ± 0.7253	6.3921 ± 0.2464	0.3196 ± 0.0393	0.3168 ± 0.0330
	ANN	92.19 ± 2.28	87.49 ± 4.02	92.063 ± 12.931	90.428 ± 17.115	0.0231 ± 0.0004	0.0234 ± 0.0006
	SVM	88.00 ± 3.56	73.36 ± 9.45	3705 ± 317.92	4050 ± 128.85	0.0055 ± 0.0006	0.0059 ± 0.0008
	LSTM	$\mathbf{95.15} \pm 2.53$	$\mathbf{92.20} \pm 4.04$	2160 ± 116.42	2302 ± 136.14	0.0153 ± 0.0006	0.0148 ± 0.0003

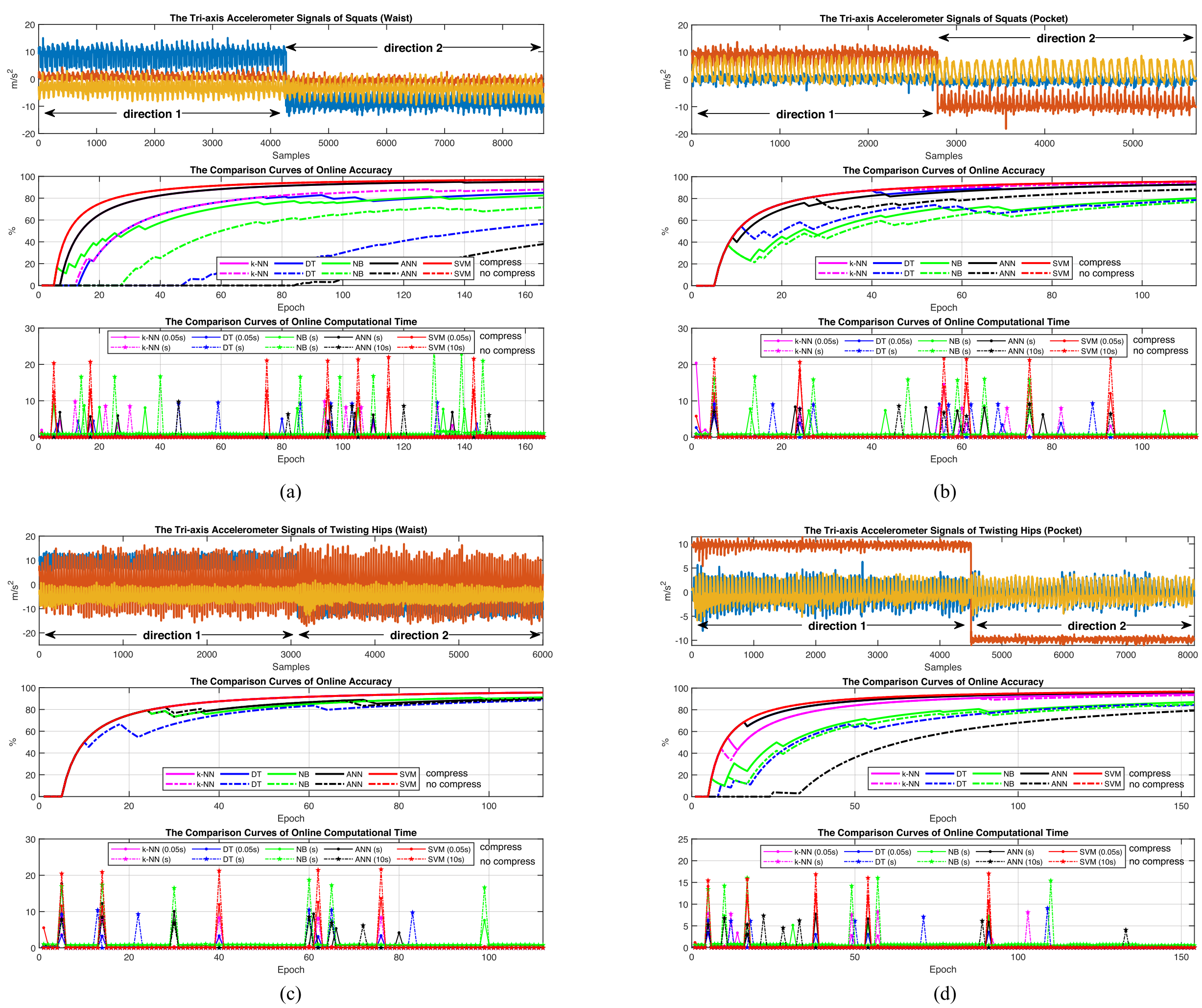

Fig. 6. Comparison of the results of predicting new activities. The three graphs are the 3-axis acceleration with two directions, the online accuracy computed by the five ML-based Ada-HAR systems and the online predicting time. The solid line and dotted line are the compressed and uncompressed datasets. The pulse points in the bottom graph are the locations at which the HC classifier was retrained. (a) Squats with the smartphone on the waist. (b) Squats with the smartphone in the pocket. (c) Twisting hips with the smartphone on the waist. (d) Twisting hips with the smartphone in the pocket.

TABLE IV
COMPARISON OF THE FINAL ACCURACY AND TOTAL TESTING TIME AMONG THE FIVE ML ALGORITHMS OF THE ADA-HAR SYSTEM

	Ada-HAR	Accuracy (%)				Total Testing Time (s)			
		Twist Hips		Squats		Twist Hips		Squats	
		Waist	Pocket	Waist	Pocket	Waist	Pocket	Waist	Pocket
Compression	k-NN	**95.54**	94.81	95.18	**95.54**	1.6699	1.7661	3.1384	4.1126
	DT	**95.54**	96.10	84.94	92.86	**1.3079**	**1.3944**	**2.7351**	**2.0972**
	NB	91.07	87.01	82.53	80.18	119.98	108.30	186.76	131.48
	ANN	90.18	96.10	95.18	92.79	65.858	34.56	49.658	53.014
	SVM	**95.54**	**96.75**	**96.99**	95.50	5.5420	5.76	7.6555	6.6831
No Compression	k-NN	**95.54**	93.51	87.95	94.59	**44.48**	44.106	67.526	**52.373**
	DT	88.39	84.42	56.63	78.38	61.672	**42.988**	**63.679**	65.842
	NB	91.07	85.06	71.69	76.58	192.37	187.23	314.95	191.79
	ANN	89.29	79.22	37.95	88.29	452.38	464.97	493.45	522.10
	SVM	**95.54**	**96.75**	**96.99**	**95.50**	1064.3	819.87	1490.4	1294.1

results of predicting twisting hips by carrying the smartphone on the waist and in the pocket. Table IV reports a comparison of the total classification accuracy and average testing time among the Ada-HAR system based on the five ML algorithms. The results prove that the compressed training dataset can save the updating time for updating the HC classifier, while k-NN and DT algorithms are the best two algorithms.

C. Demonstration

The video experiment of Ada-HAR demonstrates the recognition process in real time by changing the direction of the smartphone with two positions. The mean transmission time of local network is shorter than that of remote connection. For example, it only cost 0.18 s to receive ten new samples in the local network, while it needed 0.25 s for the remote transmission. Each result was obtained as the average of 100 repetitions.

VI. CONCLUSION

In this article, we proposed a smartphone-based adaptive recognition and real-time monitoring system (Ada-HAR) for human activities. Experiments were performed with 25 subjects to validate the performance of the developed system. The most innovative part was the online learning algorithm. It was capable of updating the classifier in a dynamic environment, i.e., which means if any new activity is found, the HC classifier will be updated automatically to include the new class. It is an unsupervised online learning algorithm that does not need to get the true labels. In addition to the adaptive algorithm, an automatic labeling method using Hk-mC algorithm was another original achievement that improves the efficiency of labeling the raw signals. The experiments proved its robustness by placing the smartphone on different positions with various directions. The introduced signal preprocessing strategy provides stable inputs for the classifier regardless of the variability of the placement of the smartphone. The results showed that the DL-based HC classifier could achieve a better performance in terms of accuracy for classifying 12 original activities (95.15% waist and 92.20% pocket). Moreover, the HC classifier based on k-NN and DT algorithms takes less time to update the previous HC classifier for identifying new classes. Compared with the state-of-the-art methods, the Ada-HAR system is an improvement not only in terms of the number of activities but also in terms of the recognition accuracy.

Future work will consider a higher number of activities and, also, even under a more complex situations as new challenges. Moreover, the developed algorithm will be validated with various wearable devices to improve its identification ability. We also expect to compare the performance of the Ada-HAR system with some commercial HAR platforms.

REFERENCES

[1] M. Kose, O. D. Incel, and C. Ersoy, "Online human activity recognition on smart phones," in *Proc. Workshop Mobile Sens.: Smartphones Wearables Big Data*, vol. 16, no. 2012, 2012, pp. 11–15.
[2] A. Aliverti, "Wearable technology: Role in respiratory health and disease," *Breathe*, vol. 13, no. 2, pp. e27–e36, 2017.
[3] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, 2018.
[4] C. Torres-Huitzil and M. Nuno-Maganda, "Robust smartphone-based human activity recognition using a tri-axial accelerometer," in *Proc. IEEE 6th Latin Amer. Symp. Circuits Syst.*, 2015, pp. 1–4.
[5] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Trans. Inf. Technol. Biomedicine*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.
[6] T. Sajana, C. S. Rani, and K. Narayana, "A survey on clustering techniques for big data mining," *Indian J. Sci. Technol.*, vol. 9, no. 3, pp. 1–12, 2016.
[7] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discovery*, vol. 22, no. 1–2, pp. 31–72, 2011.
[8] A. Wang, G. Chen, J. Yang, S. Zhao, and C.-Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4566–4578, Jun. 2016.
[9] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Statist.*, vol. 15, no. 2, pp. 265–286, 2006.
[10] S. Beura, B. Majhi, and R. Dash, "Automatic characterization of mammograms using fractal texture analysis and fast correlation based filter method," in *Proc. 2nd Int. Conf. Perception Mach. Intell.*, 2015, pp. 85–91.

[11] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[12] A. M. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, “Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis,” in *Proc. IEEE 5th Int. Conf. Future Inf. Technol. (FutureTech)*, 2010, pp. 1–6.

[13] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, “Complex human activity recognition using smartphone and wrist-worn motion sensors,” *Sensors*, vol. 16, no. 4, 2016, Art. no. 426.

[14] A. D. Ignatov and V. V. Strijov, “Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer,” *Multimedia Tools Appl.*, vol. 75, no. 12, pp. 7257–7270, 2016.

[15] W. Sousa, E. Souto, J. Rodrigres, P. Sadarc, R. Jalali, and K. El-Khatib, “A comparative analysis of the impact of features on human activity recognition with smartphone sensors,” in *Proc. 23rd Brazilian Symp. Multimedia Web*, 2017, pp. 397–404.

[16] Y.-S. Lee and S.-B. Cho, “Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data,” *Neurocomputing*, vol. 126, pp. 106–115, 2014.

[17] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-aware human activity recognition using smartphones,” *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[18] Y.-S. Lee and S.-B. Cho, “Activity recognition using hierarchical hidden Markov models on a smartphone with 3d accelerometer,” in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2011, pp. 460–467.

[19] C. A. Ronao and S.-B. Cho, “Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models,” in *Proc. IEEE 10th Int. Conf. Natural Computation*, 2014, pp. 681–686.

[20] A. Reiss, G. Hendeby, and D. Stricker, “A competitive approach for human activity recognition on smartphones,” in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, Bruges, Belgium, 2013, pp. 455–460.

[21] M. H. M. Noor, Z. Salcic, I. Kevin, and K. Wang, “Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer,” *Pervasive Mobile Comput.*, vol. 38, pp. 41–59, 2017.

[22] Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu, “Towards unsupervised physical activity recognition using smartphone accelerometers,” *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10701–10719, 2017.

[23] L. Cao, Y. Wang, B. Zhang, Q. Jin, and A. V. Vasilakos, “Gchar: An efficient group-based contextaware human activity recognition on smartphone,” *J. Parallel Distrib. Comput.*, vol. 118, pp. 67–80, 2018.

[24] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016.

[25] N. Y. Hammerla, S. Halloran, and T. Ploetz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” 2016, *arXiv:1604.08880*.

[26] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, “A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone,” *Sensors*, vol. 19, 2019, Art. no. 3731.

[27] W. Qi and A. Aliverti, “A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity,” *IEEE J. Biomed. Health Inform.*, 2019.

[28] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, “Deep learning based inference of private information using embedded sensors in smart devices,” *IEEE Netw.*, vol. 32, no. 4, pp. 8–14, Jul./Aug. 2018.

[29] O. Särkkä, T. Nieminen, S. Suuriniemi, and L. Kettunen, “A multi-position calibration method for consumer-grade accelerometers, gyroscopes, and magnetometers to field conditions,” *IEEE Sensors J.*, vol. 17, no. 11, pp. 3470–3481, Jun. 2017.

[30] J. O. Laguna, A. G. Olaya, and D. Borrajo, “A dynamic sliding window approach for activity recognition,” in *Proc. Int. Conf. User Model., Adaptation, Personalization*, 2011, pp. 219–230.

[31] P. Kanjiya, V. M. Khadkikar, and M. S. ElMoursi, “Adaptive low-pass filter based dc offset removal technique for three-phase PLLs,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 11, pp. 9025–9029, Nov. 2018.

[32] D. Roetenberg, H. J. Luinge, C. T. Baten, and P. H. Veltink, “Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 395–405, Sep. 2005.

[33] A. Rai and S. Upadhyay, “Bearing performance degradation assessment based on a combination of empirical mode decomposition and k-medoids clustering,” *Mech. Syst. Signal Process.*, vol. 93, pp. 16–29, 2017.

[34] R. Boddy and G. Smith, *Statistical Methods in Practice: For Scientists and Technologists*. Chichester, U.K.: Wiley, 2009.

[35] C. Dai, W. Chen, and Y. Zhu, “Seeker optimization algorithm for digital IIR filter design,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1710–1718, May 2009.

[36] N. M. Hewahi and M. K. Saad, “Class outliers mining: Distance-based approach.” *Int. J. Intell. Syst. Techn.*, vol. 2, no. 1, pp. 55–68, 2007.

[37] A. Liaw *et al.*, “Classification and regression by randomforest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[38] S. D. Jadhav and H. Channe, “Comparative study of k-NN, Naive Bayes and decision tree classification techniques,” *Int. J. Sci. Res.*, vol. 5, no. 1, pp. 1842–1845, 2016.

[39] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Malaysia: Pearson Education Limited, 2016.

[40] Y. Wang, X. Jia, Q. Jin, and J. Ma, “Mobile crowdsourcing: framework, challenges, and solutions,” *Concurrency Comput.: Practice Experience*, vol. 29, no. 3, 2017, Art. no. e3789.