This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

Enabling IoT for In-Home Rehabilitation: Accelerometer Signals Classification Methods for Activity and Movement Recognition

Igor Bisio, *Senior Member, IEEE,* Alessandro Delfino, Fabio Lavagetto, and Andrea Sciarrone, *Member, IEEE*

***Abstract*—Rehabilitation and elderly monitoring for active ageing can benefit from Internet of Things (IoT) capabilities in particular for in-home treatments. In this paper we consider two functions useful for such treatments: Activity Recognition (AR) and Movement Recognition (MR). The former is aimed at detecting if a patient is idle, still, walking, running, going up/down the stairs or cycling; the latter individuates specific movements often required for physical rehabilitation such as arm circles, arm presses, arm twist, curls, seaweed and shoulder rolls. Smartphones are the reference platforms being equipped with an accelerometer sensor and elements of the IoT.**

The work surveys and compares accelerometer signals classification methods to enable IoT for the aforementioned functions. The considered methods are Support Vector Machines (SVMs), Decision Trees (DTs) and Dynamic Time Warping (DTW). A comparison of the methods has been proposed to highlight their performance: all the techniques have good recognition accuracies and, among them, the SVM-based approaches show an accuracy above 90% in the case of AR and above 99% in the case of MR.

***Index Terms*—eHealth and mHealth, Activity Recognition, Movement Recognition, Accelerometer Signal Classification, Smartphones.**

I. INTRODUCTION

PHYSICAL medicine and rehabilitation (PMR), also called physiatry, is the branch of medicine emphasizing the prevention, diagnosis, and treatment of nerves, muscles, bones and brain disorders that may produce temporary or permanent impairment. Rehabilitation physicians (physiatrists) treat injuries or illnesses that affect how you move, with the aim of enhancing performance. A very important kind of PMR concerns the physical therapy for elderly people. When it comes to care for senior citizens, in-home physical therapy is one of the best options for many individuals and families. One of the most obvious benefits of in-home therapy is convenience. It can be difficult for seniors with physical conditions to travel to and from a therapists office to receive care. In-home therapy makes possible to receive professional care in the comfort of the home, allowing them to use all your energy for healing, not making travel arrangements. Moreover, outpatient physical therapy can be stressful, and offices are often noisy and crowded. In-home therapy enables to focus on each patient for longer and many patients feel that the quality of care they receive is superior: performing exercises in a stress-free environment can be more focused and intensive. The major drawback of physical therapy for seniors at home is that it is generally more expensive than outpatient care: therapy centres must reimburse their therapists for travel time and fuel costs, the cost of hiring goes up considerably and it is, naturally, transferred to patients. Fortunately, the field of home health technology is growing steadily: improved technologies will expand the options people have for in-home care. Current Information and Communications Technologies (ICTs) and, in particular, signal processing solutions may help the diffusion of in-home PMR therapy at affordable costs [1].

A person-centred modern tele-monitoring platform for PMR may be structured into four essential elements: 1) people who need rehabilitation (regarding, for example, motion and activity); 2) sensors/devices/systems measuring physical quantities, motion, and activities; 3) hubs, which collect the measurements and send them to the final destinations through telecommunication networks: they may be personal computers, laptops, and smartphones; 4) final destinations, physiatrists and other health care providers, disease management services, and family care givers.

The Internet of Things (IoT) surely offers the opportunity to implement the described platform but some specific functions must be added to make the IoT more "intelligent". Some examples of these functions can be found in the literature in [2], [3] and [4]. In the mentioned tele-monitoring platforms, smartphones can play an important role: the idea carried on in this paper, as in [5] and [6], is to give smartphone a new additional role: not only hub but "hub+sensor+processor". The idea of having a "hub+sensor" capability is widely applied [5]: sensor technologies combined with mobile communications can be used to track various health measurements for patients. Among the long list of sensors incorporated into smartphones and used for PMR monitoring there is the accelerometer that can be used to register different movements.

This paper is focused on the processing function of the smartphones and surveys accelerometer signal classification approaches, compared among each other, to perform the recognition of specific movements often required in rehabilitation protocols for both lower and upper limbs. We assume that the smartphone is with the patient (e.g., in his pocket) when monitoring typical activity related to the lower limbs (i.e., walking, running, cycling) and it is hold in the hand when an upper limb movement is performed during rehabilitation.

The approaches are suited to be implemented on a smartphone being equipped with a triaxial, piezoresistive accelerometer

The authors are with the Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, Via Opera Pia 13 16145, Genoa, Italy. E-mail:{igor.bisio, alessandro.delfino, fabio.lavagetto, andrea.sciarrone}@unige.it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

that measures the acceleration values on the three Cartesian axes. Indeed, four Android-based smartphones have been employed in this paper to collect accelerometer signal samples, for training and testing phases and to carry out the proposed performance comparisons.The accelerometer signal classification algorithms implemented over smartphones also allow avoiding the employment of specific equipment for a set of simple physical exercises often required to patients, in particular to elderly people.

The paper is organized as follows: Section II surveys the available signal classification solutions in the literature to extract movement information from the accelerometer signals. Section III presents the considered activity and movement recognition functions and the preliminary analytical definitions are provided in Section IV. The different algorithms applied in this paper are detailed in Sections V, VI and VII. Section VIII focuses on performance comparison among the presented approach in terms of accuracy. Finally, conclusions are drawn.

II. Related Work

Evaluating and tracking the daily activity of a patient is crucial for monitoring different type of patients and in recent years, the use of smartphones for home-care has gathered great interest [7]. An accelerometer-based activity detector has been used for patients undergoing pulmonary rehabilitation [8] and [9] proposed the use of a smartphone application that suggests respiration exercises, in which patients are supported remotely, and gives automatic feedback during the exercise.

The smartphone has been proved helpful for cardiac rehabilitation in postmyocardial infarction patients [10] and for monitoring and tracking progresses of heart-failure patients [5]. The accelerometer sensor has been often used to monitor patients: a step counter has been used to evaluate the daily activity of the patients [11].

Many works aim to classify the activity of the patient, by processing the accelerometer signal. Some of these works use *ad hoc* devices attached to the user body [12], but most of them exploit the in-built accelerometer of the smartphones [13]. All activity recognition algorithms require the use of machine learning solution: they extract features that represent the activity and they train a classifier with them. Many works have extracted features in the time or frequency domains and low-pass filtering is commonly used to separate the dc and ac components of an accelerometer signal [14]. Common features are mean of dc [12], [15], mean of the rectified ac signal [15], mean and standard deviation [16], [17], median and 25th and 75th percentile of a window [17], spectral energy [12], [18], entropy and correlation between axes [12]. Several works have exploited feature extracted on the wavelet transformed domain [19], [20].

Among all the possible classifiers, the most used are Artificial Neural Networks [21], [22], Naive Bayes [23], Decision Tree [23], [24], [13], k-Nearest Neighbours [24], [13], Neuro-Fuzzy [25], and Hidden Markov Model [26], [27]. [28] proposes a comparison between classifiers (Decision Tree, Bayesian Network, Naive Bayes and K-Nearest Neighbours) for activity recognition. The results of this reference show that it is possible to classify simple activities, such as walking, standing, sitting, walking up and down the stairs, with good accuracy.

On the other hand, recognizing arm movements have become very attractive for detecting gestures and interact with consumer electronics and smartphones. With the recent diffusion of smartphones, many works have been proposed to detect gestures by using a hand-held device. The detection of gestures have been performed by exploiting different classifiers, such as Hidden Markov Models [29], [30] and Support Vector Machines [30].

These methods require an extensive training phase to be effective. To overcome this problem works based on Dynamic Time Warping (DTW) have been proposed [31]. DTW was originally developed for speech recognition applications [32] but it has also been applied in many other fields such as economics [33], chemistry [34] or biology [35]. It has also been widely applied to motion detection [36], [37].

III. Recognition Functions

A. Activity Recognition

The Activity Recognition (AR) algorithm is based on processing and classification of signals provided by the smartphone-embedded accelerometer. AR is aimed at recognizing six different classes of physical activities:

1) *Idle (I)*: this class recognizes if the patient has abandoned the smartphone.
2) *Still (STI)*: represents the case in which the patient is in a no-movement condition (*e.g.,* he is sitting or standing).
3) *Walking (W)*: it is referred to the case in which the patient is walking (also on a treadmill).
4) *Running (R)*: similarly to the W class, it represent the case in which the patient is running (again, also on a treadmill).
5) *Stairs (STA)*: going up or down the stairs.
6) *Cycling (C)*: this class concerns the case where a patient is cycling by using either a bike or an exercise bike, indifferently.

The acquisition of training data for the activity recognition algorithm has been performed by keeping the smartphone in four different positions: *i)* facing toward the user; *ii)* toward the opposite side; and on the smartphone orientation: *iii)* pointing up and *iv)* pointing down [5].

B. Movement Recognition

The smartphone can also be used to check if the patient is performing one of the six movements, taken from [38], reported below. This function is called Movement Recognition (MR).

1) *Arm circles (AC)*: the patient is standing, his arms are lifted straight out to his sides at shoulder height. The patient starts moving his arms in a circular pattern (forward or backward, indifferently) with palms facing down.
2) *Arm presses (AP)*: the patient is standing with his arms along the body, his palms are facing behind him. He pushes his arms as back as he can and then he gets back to the starting position.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

3) *Arm twist (AT)*: the patient is standing, his arms are lifted straight out to his sides at shoulder height. The patient starts extending and twisting his arms. His palms move accordingly.
4) *Curls (C)*: the patient is standing, his elbows are lifted out to his sides at shoulder height, his hands are in front of his chest, palms down. The patient, without moving his elbows, draws a circle with his hands on the vertical plane, once he reaches the starting position he draws a circle backwards, getting back again to the starting position.
5) *Seaweed (SW)*: the patient is standing, his elbows are lifted out to his sides at shoulder height, his hands are in front of his chest, palms down. The patient moves his hands crossing them in front of his face and gets back to the starting position.
6) *Shoulder rolls (SR)*: the patient is standing with his arms along the body, his palms are facing the patient body. The patient starts rolling his shoulders (forward or backward, indifferently) without flexing his arms or changing the hand orientation.

IV. Preliminary Definitions

In this section we report the notation and the preliminary definitions that we use throughout the paper. For sake of readability they have also be summarized in Table I.

We denote f as the index of a frame containing N three-axis accelerometer samples $s^f_{j,n}$, which is the n-th sample, $n \in [1, N]$, of the j-th axis, $j \in \{x, y, z\}$ for the f-th frame. In the remaining of the paper, when it is not strictly necessary to distinguish among the three axes components, we omit the axis index j. Consequently, the accelerometer sample containing the three axis components $\{s^f_{x,n}, s^f_{y,n}, s^f_{z,n}\}$ is denoted with $\mathbf{s}^f_n$.

The employed features for each j-th axis are: *i)* mean (μ^f_j), *ii)* standard deviation (σ^f_j), and *iii)* number of peaks (P^f_j). Being well-known the formulae for μ^f_j and σ^f_j, we only report the definition of P^f_j:

$$P^f_j = \sum_n \rho^f_{j,n}$$
$$\rho^f_{j,n} = \begin{cases} 1, & \text{if } \left(s^f_{j,n+1} - s^f_{j,n}\right)\left(s^f_{j,n} - s^f_{j,n-1}\right) < 0,\ \|s^f_{j,n}\| \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The quantity ϵ is a threshold employed to define a signal peak, empirically set to $\epsilon = 0.75$ by of means of practical trials.

Furthermore, a kinetic indicator K^f, originally introduced in [39] and reported in Eq. (3), has been used. Such feature is, in practice, an indicator related to the energy of the accelerometer frame.

Let $i \in \{AR, MR\}$ be the recognition function label index, for the f-th frame we define $\mathbf{X}^{f,i}$ as the vector of the features defined above and $\mathbf{Y}^{f,i}$ as the vector containing all the classes of the i-th function:

$$\begin{aligned} \mathbf{Y}^{f,AR} &= \{I, STI, W, R, STA, C\} \\ \mathbf{Y}^{f,MR} &= \{AC, AP, AT, C, SW, SR\} \end{aligned} \tag{2}$$

Let $|\mathbf{Y}^{f,i}|$ be the total number of classes for each vector i and $h \in \left[1, |\mathbf{Y}^{f,i}|\right]$ be the classes' index, Y^i_h is the h-th class of the *i*-th recognition function. Given the quantities $Y^{f,i} \in \mathbf{Y}^{f,i}$ and F^i, which are the class corresponding to the vector $\boldsymbol{X}^{f,i}$ and the number of frame employed for the *i*-th recognition function, respectively, the association $\left(\mathbf{X}^{f,i}, Y^{f,i}\right), \forall f \in [1, F^i]$ is called *observation*. It represents the output of the i-th recognition function.

To lighten the notation, from hereafter we omit the index i from the equations: every time a quantity defined above will appear without the index i, it must be considered as belonging to *AR* or *MR*, indifferently.

TABLE I: The employed notation.

Symbol/ Notation	Definition
f	Frame index
n, N	Sample index, Number of accelerometer sample within a frame
j	Axis index
i	Recognition function label index
$s^f_{j,n}$	n-th accelerometric sample of the j-th axis for the f-th frame.
$\mathbf{s}^f_n$	Accelerometer sample containing the three axis components $\{s^f_{x,n}, s^f_{y,n}, s^f_{z,n}\}$
μ^f_j	Accelerometer mean for the j-th axis and for the f-th frame
σ^f_j	Accelerometer standard deviation for the j-th axis and for the f-th frame
P^f_j	Number of peaks for the j-th axis and for the f-th frame
K^f	Kinetic indicator for the f-th frame
$\mathbf{X}^{f,i}$	Feature vector for the f-th frame and the i-th recognition function
$\mathbf{Y}^{f,i}$	Classes vector for the f-th frame and the i-th recognition function
$\mathbf{Y}^{f,AR}$	Classes vector for the f-th frame and the MR recognition function
$\mathbf{Y}^{f,AR}$	Classes vector for the f-th frame and the AR recognition function
$\lvert\mathbf{Y}^{f,i}\rvert$	Number of classes of the f-th frame and the i-th recognition function
$Y^{f,i}$	Class (label) of the f-th frame and the i-th recognition function
$\hat{Y}_f$	Numeric binary labels associated to the semantic labels Y_f
F^i	Number of frame employed for the i-th recognition function

V. Support Vector Machines

The first classifiers considered in this work are Support Vector Machines (SVMs). They map inputs into a higher-dimensional feature space and then separate classes with a hyperplane. In the literature, together with Decision Tree (DT) and Dynamic Time Warping (DTW), SVM is one of the most employed classifier for activity and movement recognition. In this section, the SVM analytical model is detailed for both the AR and the MR.

As reported in [40], the most common approaches through which the SVM classifier can be employed are two: *i) One-Against-All (OAA)* and *ii) One-Against-One (OAO)*. In this paper both of them have been tested and the provided results are shown in the related section.

A. One-Against-All (OAA)

The *One-Against-All (OAA)* method builds $S_{OAA} = |\mathbf{Y}|$ SVM models, one for each Y_h class of the considered recognition function. The single SVM for the Y_h-th class is trained by employing all the feature vectors of a given training set belonging to the Y_h class with positive labels and all the other feature vectors belonging the the other classes with negative labels [40].

$$K^f = \sqrt{\frac{1}{N-1}\left[\left(\sum_n \left(s^f_{x,n}\right)^2 + \sum_n \left(s^f_{y,n}\right)^2 + \sum_n \left(s^f_{z,n}\right)^2\right) - \frac{1}{N}\left(\left(\sum_n s^f_{x,n}\right)^2 + \left(\sum_n s^f_{y,n}\right)^2 + \left(\sum_n s^f_{z,n}\right)^2\right)\right]} \quad (3)$$

Starting from the accelerometer signals of the training set, the single SVM, built for the class Y_h, can be obtained by computing the aforementioned hyperplane that can be expressed as a function of its orthogonal vector $\mathbf{w}$ that can be obtained from the *Lagrangian Multipliers* computed by solving the problem below.

$$\begin{aligned} &\underset{Y_h, h\in[1,|\mathbf{Y}|]}{\mathbf{SVM}^{OAA}} : \min_{\boldsymbol{\lambda}} \boldsymbol{\Gamma}(\boldsymbol{\lambda}) = \\ &\frac{1}{2}\sum_{f_1=1}^{F}\sum_{f_2=1}^{F} \hat{Y}_{f_1}\hat{Y}_{f_2}\cdot\phi\left(\boldsymbol{X}_{f_1}, \boldsymbol{X}_{f_2}\right)\lambda_{f_1}\lambda_{f_2} + \\ &\qquad - \sum_{f_1=1}^{F}\lambda_{f_1}, \\ &\sum_{f_1=1}^{F}\lambda_{f_1}\hat{Y}_{f_1} = 0, \\ &0 \le \lambda_{f_1} \le C, \ \forall f_1. \end{aligned} \quad (4)$$

$\boldsymbol{\lambda}^Y = \{\lambda^Y_1 \cdots \lambda^Y_{f_1}, \lambda^Y_{f_2} \cdots \lambda^Y_F\}$ represents the *Lagrangian Multipliers* vector for $\mathbf{Y}$ previously mentioned and the quantity C $(C > 0)$ is the *Complexity* constant [41].
The scalars $\hat{Y}_f \in [-1, 1]$ are the numeric binary labels associated to the semantic labels Y_f of the feature vectors $\boldsymbol{X}_f$. Obviously, in this phase, the feature vector is referred to a training set of accelerometer signals acquired and suitably labelled *a priori*. They are defined as:

$$\hat{Y}_f = \begin{cases} 1, & \text{if } Y_f \equiv Y_h \\ -1, & \text{otherwise} \end{cases} \quad (5)$$

Eq. (4) represents a non-linear SVM. It contains the function $\phi\left(\boldsymbol{X}_{f_1}, \boldsymbol{X}_{f_2}\right)$ which is the *Radial Basis Function* (RBF) or *Gaussian Kernel* in this paper:

$$\phi\left(\boldsymbol{X}_{f_1}, \boldsymbol{X}_{f_2}\right) = e^{-\gamma||\boldsymbol{X}_{f_1} - \boldsymbol{X}_{f_2}||} \quad , \gamma > 0 \quad (6)$$

where the quantity γ is a RBF Kernel parameter [41].
The problem (one for each class $Y_h, h \in [1, |\mathbf{Y}|]$) in Eq. (4) is solved with the *Sequential Minimal Optimization* (SMO) by following the guidelines reported in [42] and the relative library available in [43].

B. One-Against-One (OAO)

The *One-Against-One (OAO)* method relies on constructing classifiers where each one is trained by using data from two classes. Differently from the *OAA* approach, the *OAO* method aims at training different SVMs by using combinations (i.e., the order of selection does not matter) of all the classes $|\mathbf{Y}|$ in groups of two. This leads to obtain $S_{OAO} = \binom{|\mathbf{Y}|}{2} = \frac{|\mathbf{Y}|\cdot(|\mathbf{Y}|-1)}{2}$ different SVMs. Each single SVM can be expressed as:

$$\underset{\substack{Y_h, Y_k \\ h,k\in[1,|\mathbf{Y}|] \\ h\neq k}}{\mathbf{SVM}^{OAO}} : \min_{\boldsymbol{\lambda}} \boldsymbol{\Gamma}(\boldsymbol{\lambda}) \quad (7)$$

In this case the number of considered frames is $F_h + F_k$ where F_h and F_k are the number of frames belonging to the h-th and k-th classes, respectively. The numeric binary labels $\hat{Y}_f$ are computed as follow:

$$\hat{Y}_f = \begin{cases} 1, & \text{if } Y_f \equiv Y_h \\ -1, & \text{if } Y_f \equiv Y_k \end{cases} \quad (8)$$

Again, SVMs are trained as in [43].

C. Decision Rules

Remembering that F is the number frames (and feature vectors) that compose the accelerometer signal acquired by the patient's smartphone and defining $|\mathbf{X}_f|$ as the number of features for each frame, the $F \times |\mathbf{X}_f|$ matrix $\boldsymbol{\omega}$, containing all the feature vectors is:

$$\boldsymbol{\omega} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_f \\ \vdots \\ \mathbf{X}_F \end{bmatrix} \quad (9)$$

Every time that an accelerometer signal must be classified, the SVM outputs a $F \times 2$ matrix $\boldsymbol{\Omega}\{\boldsymbol{\omega}\}$, called *Probability Matrix* (PM), independently of the approach considered (*OAA* rather than *OAO*).

Each element Ω_{ft} with $f \in [1, F]$ and $t \in [1, 2]$ of this matrix is the the *a-posteriori* probability of the f-th feature vector $\mathbf{X}_f$ to belong to the t-th class, as reported in Eq.(10).

$$\boldsymbol{\Omega}\{\boldsymbol{\omega}\} = \begin{bmatrix} Pr\left(\hat{Y}_1 = 1|\mathbf{X}_1\right) & Pr\left(\hat{Y}_1 = -1|\mathbf{X}_1\right) \\ \vdots & \vdots \\ Pr\left(\hat{Y}_F = 1|\mathbf{X}_F\right) & Pr\left(\hat{Y}_F = -1|\mathbf{X}_F\right) \end{bmatrix} \quad (10)$$

1) OAA Decision Rule: Denoting with $\boldsymbol{\Omega}^h\{\boldsymbol{\omega}\}$ the PM associated to the SVM of the h-th class Y_h and Ω^h_{ft} its element (see Eq.(4)), the decision rule for the *OAA* approach, providing the index h^* of the predicted class $Y_{h^*} \in \mathbf{Y}$, is the following:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

$$h^* = \arg\max_h \left[\frac{1}{F} \sum_{f=1}^{F} \Omega_{f1}^{h} \right] \quad \forall h,\ h \in [1, |\mathbf{Y}|] \tag{11}$$

This approach, similar to the "*Max Wins*" proposed in [40], aims at classifying the accelometer signal, starting from its feature vectors $\mathbf{X}_f$ ($\forall f,\ f \in [1, F]$), as belonging to the class that has the maximum average probability.

2) OAO Decision Rule: Similarly to the *OAA* decision rule, we denote with $\mathbf{\Omega}^{hk}\{\boldsymbol{\omega}\}$ the PM associated to the SVM for the h-th and k-th classes Y_h and Y_k. As a consequence, the decision rule for the *OAO* case becomes:

$$h^* = \arg\max_h \left[\frac{1}{F \cdot (|\mathbf{Y}| - 1)} \sum_{\substack{h=1 \\ h \neq k}}^{|\mathbf{Y}|-1} \sum_{f=1}^{F} \Omega_{f1}^{hk} \right] \quad \forall h,\ h \in [1, |\mathbf{Y}|] \tag{12}$$

VI. C4.5 Decision Tree

Another classifier compared in this paper is the Decision Tree (DT). The classifiers of this family are non-parametric supervised learning algorithms employed both for classification and regression. Also DTs are based on a training phase aimed at creating a model that recursively partitions the feature space such that the accelerometer signals with the same labels (i.e., representing the same movement/activity) are grouped together.

A DT can be represented as a flowchart-like structure in which each node represents a test (i.e., comparison with respect to the threshold δ defined below) on a single feature of the considered feature vector, each branch represents the outcome of the test and each leaf (or terminal node) represents a class label. In practice, the paths from root (i.e., the first node) to leaf represents the classification rules defined during the training phase. The bigger the tree, the more complex the decision rules and the higher the risk of over-fitting. On the other hand, a smaller tree provides simpler decision rules and a more generic DT but with lower classification accuracy. The DT employed in this paper is the C4.5.

A. C4.5 Algorithm Description

The implemented version of the algorithm, employed in this paper, is based on the so called *split*. It is an association $\theta(x, \delta)$ where x is the index of the x-th component of the features vector of the f-th frame $\mathbf{X}_f$ (i.e., it is a single feature for a given frame) of the accelerometer signals belonging to the training set denoted, again, with $\boldsymbol{\omega}$.

δ is the threshold employed to separate the overall training set into two subsets on the basis of the value assumed by the x-th feature. δ is the threshold also employed for the test once the DT is trained.

In particular, the subset $\boldsymbol{\omega}_{left}(\theta)$ represents the subset of signals whose x-th feature has a value below (or equal) the threshold δ. Vice versa, the signals having the x-th feature above δ belongs to the subset $\boldsymbol{\omega}_{right}(\theta)$.

The *split* θ must be carefully chosen by selecting a proper criterion that allows deciding which x-th feature must be used and with which threshold δ. It can be done by defining the so called *purity* function $G(\theta)$ that represents a measure of how well the classes are separated (see [44] for details).

$$G(\theta) = \frac{|\boldsymbol{\omega}_{left}|}{|\boldsymbol{\omega}_{left}| + |\boldsymbol{\omega}_{right}|} \cdot H_{left}(\theta) + \frac{|\boldsymbol{\omega}_{right}|}{|\boldsymbol{\omega}_{left}| + |\boldsymbol{\omega}_{right}|} \cdot H_{right}(\theta) \tag{13}$$

$H_{left}(\theta)$ is called left-entropy function:

$$H_{left}(\theta) = \sum_{h=1}^{|\mathbf{Y}|} \frac{\omega_{left}^{h}(\theta)}{|\boldsymbol{\omega}_{left}(\theta)|} \log_2 \frac{\omega_{left}^{h}(\theta)}{|\boldsymbol{\omega}_{left}(\theta)|} \tag{14}$$

where $\omega_{left}^{h}(\theta)$ is the number of element of the set $\boldsymbol{\omega}_{left}(\theta)$ belonging to the h-th class of the vector $\mathbf{Y}$. Equivalently, $H_{right}(\theta)$ is called right-entropy function where, obviously, $\omega_{right}^{h}(\theta)$ is the number of element of the set $\boldsymbol{\omega}_{right}(\theta)$ belonging to the h-th class of the vector $\mathbf{Y}$:

$$H_{right}(\theta) = \sum_{h=1}^{|\mathbf{Y}|} \frac{\omega_{right}^{h}(\theta)}{|\boldsymbol{\omega}_{right}(\theta)|} \log_2 \frac{\omega_{right}^{h}(\theta)}{|\boldsymbol{\omega}_{right}(\theta)|} \tag{15}$$

High values of $G(\theta)$ represent a bad separation between the classes while low values mean that the *split* is able to properly separate the classes. The metric for deciding the best *split* is the *Information Gain*, defined as:

$$IG(\theta) = H - G(\theta) \tag{16}$$

where H is the entropy before the *split* and is defined as:

$$H = \sum_{h=1}^{|\mathbf{Y}|} \frac{\omega^h}{|\boldsymbol{\omega}|} \log_2 \frac{\omega^h}{|\boldsymbol{\omega}|} \tag{17}$$

in which ω^h is the number of element, belonging to the h-th class of the vector $\mathbf{Y}$, of the set $\boldsymbol{\omega}$, which contains all the considered accelerometer signals before the *split*.

The choice of the best *split* falls on the value of θ which provides the best result in terms of maximizing the *Information Gain*. Analytically:

$$\theta^* = \arg\max_\theta [IG(\theta)] \tag{18}$$

The C4.5 DT algorithm iterates until the maximum allowable depth (i.e., the maximum number of *splits* from the root to the leafs of the DT) is reached or a stop condition is satisfied. In this paper the employed stop condition is that a terminal node (i.e., a *leaf* of the DT) contains less samples than a predefined threshold (called *MinNumberObject* often expressed in percentage). This stop modality is also called

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

on-line pruning of the DT [45].
Finally, once the DT has been inducted, starting from the training set, some *leaf* can be pruned (*post-pruning*) to reduce the size of the DT (i.e., the number of nodes) to avoid unnecessary complexity. This final pruning is carried out by pruning nodes whose cancellation does not significantly impact the overall classification error. The parameter used to tune the *post-pruning* is the *Confidence Factor*: lowering the *Confidence Factor* decreases the amount of *post-pruning*. In this work we limit the amount of pruning and we set the *Confidence Factor* equal to 0.01, as reported in the performance comparison.

VII. Dynamic Time Warping

Differently from the previous considered approaches, all based on features extracted from the accelerometer signals, the Dynamic Time Warping (DTW) [46] is a technique that finds an optimal alignment between two sequences (i.e., the accelerometer signals samples). It does not uses features to classify signals but finds a non linear mapping between two signals and retrieves a distance measure. It is worth noticing that in the case of DTW the signal samples are not referred to a frame but to the entire signal.

A. DTW Description

We denote with $\mathbf{S^A} = \{\mathbf{s}_1^A, ..., \mathbf{s}_n^A, ..., \mathbf{s}_N^A\}$ and $\mathbf{S^B} = \{\mathbf{s}_1^B, ..., \mathbf{s}_m^B, ..., \mathbf{s}_M^B\}$ the two signals to be compared, with $n \in [1, N]$ and $m \in [1, M]$.
To compare two values $\mathbf{s}_n^A$ and $\mathbf{s}_m^B$ a the *DTW* algorithm needs a local cost function $c(\mathbf{s}_n^A, \mathbf{s}_m^B)$ that, given two triplets of accelerometric signal $\mathbf{s}_n^A$ and $\mathbf{s}_m^B$, retrieves a positive number that is small if $\mathbf{s}_n^A$ and $\mathbf{s}_m^B$ are similar and is large otherwise. In our implementation the cost function is chosen to be the 2-norm of the difference between $\mathbf{s}_n^A$ and $\mathbf{s}_m^B$, as reported in Eq. (19).

$$c(\mathbf{s}_n^A, \mathbf{s}_m^B) = \|\mathbf{s}_n^A - \mathbf{s}_m^B\|_2 \tag{19}$$

The cost measure $c(\mathbf{s}_n^A, \mathbf{s}_m^B)$ is symmetric, so $c(\mathbf{s}_n^A, \mathbf{s}_m^B) = c(\mathbf{s}_m^B, \mathbf{s}_n^A)$. Computing the local costs for each pair of elements of $\mathbf{S^A}$ and $\mathbf{S^B}$ the $N \times M$ elements of the cost matrix $\mathbf{C}$ is obtained as in Eq. (20).

$$C_{n,m} = c(s_n^A, s_m^B), \ \forall n \in [1, N], \forall m \in [1, M] \tag{20}$$

A warping path, within the matrix $\mathbf{C}$, can be defined as $\mathbf{P} = (p_1, ..., p_l, ..., p_L)$ where L is the total path length and $p_l = (n_l, m_l)$ for $l \in [1, L]$ is the l-th pair of indexes (n, m) of the matrix $\mathbf{C}$. Such warping path must satisfy the following three conditions:

1) Boundary conditions:
$p_1 = (1, 1)$ and $p_L = (N, M)$.
2) Continuity:
$p_l - p_{l-1} \in \{(0, 1), (1, 0), (1, 1), \forall l \in [2, L]\}$. This limits the step size.
3) Monotonicity:
$n_1 \leq n_2 \leq ... \leq n_N$ and $m_1 \leq m_2 \leq ... \leq m_M$. This forces the points in $\mathbf{P}$ to be monotonically spaced in time.

The goal of the considered technique is to find the optimal warping path $\mathbf{P}^*$ between $\mathbf{S^A}$ and $\mathbf{S^B}$ having the minimal overall cost among all the possible paths. The cost of the warping path is defined as:

$$Cost_{\mathbf{P}}(\mathbf{S^A}, \mathbf{S^B}) = \sum_{l=1}^{L} c(\mathbf{s}_{n_l}^A, \mathbf{s}_{m_l}^B) \tag{21}$$

The DTW distance between the two sequences $\mathbf{S^A}$ and $\mathbf{S^B}$ is defined as the overall cost of the optimal path $\mathbf{P}^*$:

$$DTW(\mathbf{S^A}, \mathbf{S^B}) = Cost_{\mathbf{P^*}}(\mathbf{S^A}, \mathbf{S^B}) \tag{22}$$

B. DTW Implementation

To find the optimal path $\mathbf{P}^*$ the brute force approach is often impractical. It would mean to test every possible warping path between $\mathbf{S^A}$ and $\mathbf{S^B}$, and would have a computational complexity that is exponential with respect to the number of samples of the two compared accelerometer signals, N and M. For this reason, in this paper an algorithm based on dynamic programming has been used, coherently with the literature [46], to compute the DTW distance. The algorithm uses the so called accumulated cost matrix $\mathbf{D}$ that is defined as follows:

$$D_{a,b} = DTW(\mathbf{S^A}, \mathbf{S^B}), \quad n \in [1, a], m \in [1, b] \tag{23}$$

Starting with $D_{0,0} = 0$, $D_{a,b}$ can be computed iteratively:

$$\begin{gathered} D_{a+1,b+1} = c(s_a^A, s_b^B) + \\ + \min\left(D_{a+1,b+1}, D_{a,b+1}, D_{a+1,b}\right) \end{gathered} \tag{24}$$

A local constraint is added to limit the search for the match between $\mathbf{S^A}$ and $\mathbf{S^B}$:

$$|a - b| \leq W \tag{25}$$

The value of W is a parameter of the algorithm. It has been set, after some trials, equal to 50 as reported in the next section. It is finally worth noting that DTW algorithm is robust with respect to the length of sequences: DTW distance does not significantly change if N and M have not the same value.

C. Decision Rule

To apply the DTW technique to the activity/movement classification task, a decision rule to select the appropriate class is needed. In this paper, the following rule has been implemented and employed. For each class belonging to the class set $\mathbf{Y}$ a Reference Signal (RS) is chosen. The RS, denoted with $\mathbf{R}^h = \{r_1^h, \ldots, r_n^h, \ldots, r_N^h\}$, $n \in [1, N]$, is the RS for the class Y_h and it is used for comparison.
When an unlabelled accelerometer signal, denoted with $\mathbf{U} = \{u_1, \ldots, u_m, \ldots, u_M\}$, $m \in [1, M]$ is classified, the employed decision rule is based on the comparison of $\mathbf{U}$ with all the RSs by measuring the *DTW* distances between them. The decision (i.e., the selected activity/movement class) h^* is the class whose RS is DTW-closer to $\mathbf{U}$:

$$h^* = \arg\min_h \left(\mathbf{R}^h, \mathbf{U}\right) \tag{26}$$

TABLE II: High-level comparison of the algorithms employed in this paper with other solutions

	Recognized Movements	Involved Body Part		*Ad-hoc* sensor needed	Additional HW	Accuracy	Easiness of use
		Upper limbs	Lower limbs				
[3]	Rehabilitative / Daily	X	X	Yes	Yes	High	Low
[12]	Daily	X	X	Yes	Yes	Average	Average
[13]	Daily	X	X	Yes	None	Average	High
[14]	Daily		X	Yes	Yes	High	High
[17]	Daily		X	Yes	Yes	Average	Low
[21]	Daily		X	None	None	High	High
[22]	Rehabilitative / Daily		X	Yes	Yes	High	Average
[25]	Daily		X	Yes	None	Very high	High
[31]	Gesture	X		Yes	None	Very high	High
Our paper	**Rehabilitative / Daily**	**X**	**X**	**None**	**None**	**Very high**	**High**

VIII. Performance Comparison

The set of accelerometer signals employed in this paper for the comparison of the surveyed techniques has been acquired by using three different smartphones: *i)* Samsung Galaxy S4, *ii)* Samsung Galaxy Note 3 and *iii)* Nexus 5. An *ad-hoc* Android application has been developed and installed over the considered mobile devices so to acquire signals. It is important pointing out that the conducted analysis does not depend on the particular smartphones used since changing the reference mobile device does not influence the performance of the proposed solutions. The employed smartphones integrates a triaxial, piezoresistive accelerometer that measures the acceleration values on the three Cartesian axes in $\left[\frac{m}{s^2}\right]$. The collected raw measurements (one for each Cartesian axis: x, y, z) have been stored and employed for the performance comparison detailed below. When the evaluated approach requires the framing of the signals, the frame duration has been set equal to $4\ s$. For the Activity Recognition (AR) case a set of about 14 hours has been employed. Acquisitions were performed by 8 users who kept the smartphones in four different positions and orientations: *a)* facing towards the user, *b)* towards the opposite side, *c)* pointing up, *d)* pointing down. For the Movement Recognition (MR) case the employed set consists of about 1500 accelerometer signals (corresponding to about 2 hours). Such signals have been acquired by 6 users who have kept the smartphones in their right hand when they performed the movements, independently of the devices orientation. The whole database (AR and MR), already exported in Matlab environment, is downloadable [47] and available for possible further experiments and comparisons of the surveyed approaches or other possible alternatives.

A. Comparison with other solutions

Table II shows an high-level comparison of the algorithms employed in this paper with other solutions present in the literature. For sake of readability, the first column of the table reports the typology of movements simply divided into two categories: *i)* rehabilitative, if the movements are aimed at recover a patent's injury or *ii)* daily, if they aim at understand which kind of activity a user is performing during the day. State of the art solutions are further divided among those who monitor lower limbs movements, upper limbs movements or both of them. In addition some algorithms require *ad-hoc* sensors (wearable or specifically built) or additional hardware (to complete the system they propose). Finally, the accuracy and the easiness of use are considered.

As Table II highlights, many state of the art solutions involved lower limbs movements recognitions. Almost all of them employ *ad-hoc* sensors and many of them require addition hardware (such as laptop PC). The accuracy is often high as well as the easiness of use.

Differently from the other approaches the proposed paper is able to monitor both lower and upper limb movements, both for rehabilitative and daily aims, without the need of *ad-hoc* sensors rather than additional hardware since only an off-the-shelf smartphone is required. Furthermore, our work reaches very high accuracy keeping the easiness of use very high.

B. Activity Recognition

1) AR with SVM: the two SVM approaches, *OAA* and *OAO*, have been tested. Both the approaches employ SVMs that have been trained by using around 10 minutes of accelerometer signals for each considered class. The experiments have been performed with signals not used in the training phase. In more detail, a 10-fold cross-validation approach has been employed for both, OAA and OAO, SVMs.

a) SVM-OAA: the results of the *One-Against-All (OAA)* approach are shown in the Confusion Matrix (CM) reported in Table III where rows represent the actual class while columns represent the output of the considered classifier. The elements of the CM are the percentages of frames of the class indicated in a row classified as frames belonging the class indicated in the columns. The overall accuracy, i.e., the average of the classification accuracies reported in the CM diagonal, is 75%. The SVM parameters have been reported in caption of the table.

TABLE III: AR Confusion Matrix for the SVM-*OAA* case. $C = 1$, $\gamma = 0.1$. Overall Accuracy: 75%.

	I	STI	W	R	STA	C
I	100	0	0	0	0	0
STI	21	79	0	0	0	0
W	0	0	62	2	33	0
R	0	0	0	67	30	3
STA	0	0	7	13	68	12
C	0	0	0	0	28	72

b) SVM-OAO: the numerical results yielded by the *One-Against-All (OAO)* approach are shown in the CM reported in Table IV. Again the table's caption reports the values of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

the SVM parameters. The SVM-OAO exhibits a good overall accuracy, above 90%.

TABLE IV: AR Confusion Matrix for the SVM-*OAO* case. $C = 1$, $\gamma = 0.1$. Overall Accuracy: 91%.

	I	STI	W	R	STA	C
I	100	0	0	0	0	0
STI	9	91	0	0	0	0
W	0	0	89	0	11	0
R	0	0	0	92	8	0
STA	0	0	7	3	90	0
C	0	0	0	0	19	81

2) DT-C4.5: in the same way of the SVM cases, C4.5 DT has been trained by using features extracted from 10 minutes of accelerometer signals for each class of movement. Again, in order to fairly test this classifier with respect to the other, a 10-fold cross validation has been employed. The results are reported in Table V where the salient parameters of the DT employed have been reported. In particular, the C4.5 training approach is ruled by 2 parameters: the *ConfidenceFactor* and the *MinNumberObj* per leaf. As already mentioned previously, the former is a pruning-related parameter while the latter is a parameter related to the minimum number of instances a left must classifies to be considered as so. After a tuning phase the *ConfidenceFactor* has been set to 0.01 and the *MinNumberObj* per leaf has been set to the 5% of the total number of instances. Compared to the SVM-OAO case, C4.5 DT provides lower accuracy.

TABLE V: AR Confusion Matrix for the C4.5 DT. *ConfidenceFactor*=0.01, *MinNumberObject*=5%. Overall Accuracy: 84%.

	I	STI	W	R	STA	C
I	99	1	0	0	0	0
STI	9	91	0	0	0	0
W	0	0	54	5	34	7
R	0	0	1	98	1	0
STA	0	0	8	0	89	3
C	0	0	15	1	12	72

3) DTW: for each class one signal of the acquired set has been set as RS while the remaining instances are used for classification (i.e., it has not been used the 10-fold approach). The results of the classification are summarized in the CM reported in Table VI. The performance, in terms of correct decisions, is poor: 61%. This is due to the fact that DTW tries to find similar patterns regardless the speed or duration of the activity/movement. *Walking* and *Running*, for example have quite similar accelerometer patterns, while they differ in speed.

4) Comparison between Classifiers: The evaluation is done by using the following metrics: *True Positive Rate (TPR)*, *False Positive Rate (FPR)*, *Precision (P)*, *Recall (R)*, and *F-Score (FS)*. They have been computed as reported in [48] and defined

TABLE VI: AR Confusion Matrix for the DTW. $W = 50$, Overall Accuracy: 61%.

	I	STI	W	R	STA	C
I	100	0	0	0	0	0
STI	0	80	1	0	18	2
W	0	0	51	0	49	0
R	0	0	1	43	56	0
STA	0	0	1	24	75	0
C	19	1	7	8	49	16

TABLE VII: A comparison of classifiers proposed for AR.

Classifier	TPR	FPR	P	R	FS
SVM OAO $C = 1, \gamma = 0.1$	90.50	1.90	91.93	90.50	91.21
SVM OAA $C = 1, \gamma = 0.1$	75.04	5.01	79.90	75.04	77.39
C4.5 DT *MinNumObj* = 10% *ConfidenceFactor* = 0.1	84	3.2	84.69	84	84.34
DTW $W = 50$	60.82	7.83	74.01	60.82	66.77

here for completeness.

For the h-th class, we define the correctness of a classification a the number of correctly recognized frames of such a class (true positives, tp_h), the number of correctly recognized frames that do not belong to the class (true negatives, tn_h) and the frames that either were incorrectly assigned to the class (false positives, fp_h) or that were not recognized as frames of such a class (false negatives, fn_h). Starting from this definitions, the aforementioned metrics are:

- *True Positive Rate (TPR)*:

$$TPR = \frac{1}{|\mathbf{Y}|}\sum_{h=1}^{|\mathbf{Y}|} tp_h \tag{27}$$

- *False Positive Rate (FPR)*:

$$FPR = \frac{1}{|\mathbf{Y}|}\sum_{h=1}^{|\mathbf{Y}|} fp_h \tag{28}$$

- *Precision (P)*:

$$P = \frac{1}{|\mathbf{Y}|}\sum_{h=1}^{|\mathbf{Y}|} \frac{tp_h}{tp_h + fp_h} \tag{29}$$

- *Recall (R)*:

$$R = \frac{1}{|\mathbf{Y}|}\sum_{h=1}^{|\mathbf{Y}|} \frac{tp_h}{tp_h + fn_h} \tag{30}$$

- *F-Score (FS)*:

$$FS = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot (P + R)} \tag{31}$$

In this paper we considered $\beta = 2$.

As Table VII shows, the SVM *OAO* is the classifier which provides the best result. It has the highest *TPR*, *P*, *R* and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2016.2628938, IEEE Internet of Things Journal

TABLE VIII: MR Confusion Matrix for the SVM-OAA classifier. $C = 1$, $\gamma = 0.1$. Overall Accuracy: 93.7%.

	AC	AP	AT	C	SW	SR
AC	97	3	0	0	0	0
AP	0	100	0	0	0	0
AT	0	0	97	3	0	0
C	0	0	0	100	0	0
SW	0	0	30	0	70	0
SR	0	0	0	0	0	100

TABLE IX: MR Confusion Matrix for the SVM-OAO classifier. $C = 1$, $\gamma = 0.1$.
Overall Accuracy: 99.3%.

	AC	AP	AT	C	SW	SR
AC	100	0	0	0	0	0
AP	0	100	0	0	0	0
AT	0	0	100	0	0	0
C	0	0	0	100	0	0
SW	0	0	4	0	96	0
SR	0	0	0	0	0	100

FS. At the same time it also presents the lowest *FPR*. This can be motivated by the fact the the *OAO* approach requires $\frac{|\mathbf{Y}|\cdot(|\mathbf{Y}|-1)}{2}$ different SVMs with respect to the *OAA* which only needs $|\mathbf{Y}|$. The DT also exhibits a good performance with a *TPR* of 84% and a *FPR* lower than 4%.
Finally, the *DTW* algorithm is the one showing the worst results: *TPR* and *FPR* around 60% and 7%, respectively. These results find their explanation in the consideration that the *DTW* has been used for recognizing activities that are characterized by very similar movements and simply differ one from another in terms of the speed with which they are performed (for example, consider *Walking* with *Running* and *Stairs*). Moreover, the DTW algorithm is very sensitive to the phone orientation. For these reasons the DTW algorithm, for the AR case, often yields incorrect classifications.

C. Movement Recognition

Also in the MR case, the technique of 10-fold cross-validation has been used for testing all the considered approaches, except DTW, for the reason explained before.

1) SVM: we test the two SVM training approaches: *One-Against-All (OAA)* and *One-Against-One (OAO)*.

a) SVM-OAA: the *OAA* approach requires to train less SVMs, only 1 for each class. Therefore the overall performance is poorer with respect to the SVM-*OAO* although it is still good. The accuracy is 93.7%. This is due mainly by the *Seaweed* correct detections (70%) as shown by the CM reported in Table VIII.

b) SVM-OAO: the *OAO* approach is more complex since 15 SVM classifiers had to be trained, one for each combination of 2 classes. This approach is the one that gives the best performance: the overall accuracy is 99.3%. It can be seen in Table IX reporting the CM. It only fails few times by labelling as *Arm Twist* the 4% of the *Seaweed* movements.

2) DT-C4.5: In this case, again after a tuning phase as done for the AR, the *ConfidenceFactor* has been set to 0.01 (as previously) and the *MinNumberObj* per leaf has been set to the 20% of the total number of instances. The performance is poorer than the SVM classifiers, the DT shows an overall accuracy of 76.8%. The classification results are summarized by the CM in Table X.

TABLE X: MR Confusion Matrix for the DT classifier. *ConfidenceFactor*=0.01, *MinNumberObject*=20%. Overall Accuracy: 76.8%

	AC	AP	AT	C	SW	SR
AC	75.5	0	6.5	13	5.5	0
AP	0	89	6	6	0	0
AT	15.5	5	49	0	31	0
C	14	0	5	71	10	0
SW	0	0	14	0	86	0
SR	0	9.5	0	0	0	90.5

3) DTW: as previously said, it is not possible to use the 10-fold cross-validation to evaluate the DTW since it needs one Reference Sequence (RS) for each class. Therefore, as done for AR, for each class one instance has been selected as RS and all the other instances have been used for testing. The results of the tests have been reported in Table XI. The overall accuracy is good although it is still worse than the SVMs'.

TABLE XI: MR Confusion Matrix for the DTW classifier. *W*=50. Overall Accuracy: 88.8%

	AC	AP	AT	C	SW	SR
AC	99	0	0	0	0	1
AP	0	75	0	0	0	25
AT	0	0	94	0	0	6
C	25	0	0	66	0	9
SW	2	0	0	0	98	0
SR	0	0	0	0	0	100

4) Comparison between Classifiers: The evaluation is effectuated by using the same metrics: *TPR*, *FPR*, *Precision (P)*, *Recall (R)*, and *F-Score (FS)*. Table XII shows that, again, the SVM is the classifier which provides the best result. It has the highest *TPR*, *P*, *R* and *FS*. As a consequence, it presents the lowest *FPR*. On the other hand, differently from the AR case, the *DTW* provides good results (with a *TPR* around 90%). This

TABLE XII: A comparison of classifiers proposed for MR.

Classifier	TPR	FPR	P	R	FS
SVM OAO $C = 1, \gamma = 0.1$	99.33	0.13	99.36	99.33	99.35
SVM OAA $C = 1, \gamma = 0.1$	94.54	1.20	95.09	94.00	94.54
C4.5 DT *MinNumObj* = 10% *ConfidenceFactor* = 0.1	76.83	4.68	77.09	76.58	76.84
DTW $W = 50$	88.8	2.27	91.58	88.67	90.10

is due to the fact that every movement considered significantly differs from the other, so helping the algorithm having a better match between the accelerometer signals. DT shows the worst result.

D. Single Movement Fine Classification

ArmCircles (AC) class can be divided in *ArmCirclesFront* (ACF), in which the movement is performed forward, and in *ArmCirclesBack* (ACB) in which it is performed backward. Table XIII and Figure 1 report a performance comparison when these two classes are considered separated, with respect to *SVM-OAO* and *DTW* algorithms. Table XIII shows that the *SVM-OAO* classifier is not able to distinguish the movement orientation since *ACB* is always recognized. On the other hand, *DTW* can separate the two classes even if the performances are not so good. Figure 1 provides a more detailed analysis. The *DTW* exhibits a *TPR* around 70% while the *SVM* is 50%; the *FPR* is lower than 30% for the *DTW* while is again 50% for the SVM. Similar considerations also hold for *Precision (P)*, *Recall (R)*, and *FS*.

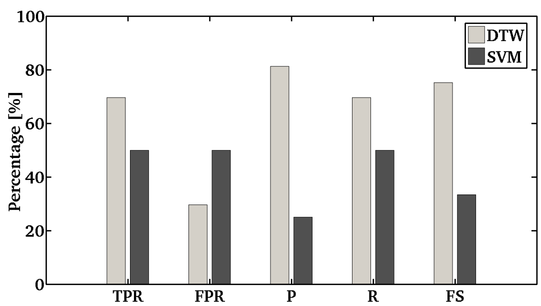
TABLE XIII: ACF and ACB accuracies comparison.

Percentages with *SVM-OAO* (left) and with *DTW* (right).

	ACF	**ACB**
ACF	0	100
ACB	0	100

	ACF	**ACB**
ACF	40	60
ACB	2.28	97.72

Fig. 1: Comparison between SVM and DTW for AC case.



In the same way, the *ShouldersRolls* (SR) class can be split in *ShouldersRollsFront* (SRF), forward movement, and in *ShouldersRollsBack* (SRB), backward movement. Table XIV and Figure 2 report a performance comparison when these two classes are considered separated, with respect to *SVM* and *DTW* algorithms. Similarly to the AC case, Table XIV shows that the *SVM* classifier cannot distinguish the movement orientation since *SRF* is always recognized. Again, *DTW* is able to separate the two classes even though some errors occur. Figure 2 exhibits a graphical analysis. The *DTW* shows a *TPR* around 80% while the *SVM* is 50%; the *FPR* is near to 20% for the *DTW* while is again 50% for the SVM. As already commented for the AC case, similar considerations also hold for *Precision (P)*, *Recall (R)*, and *FS*.

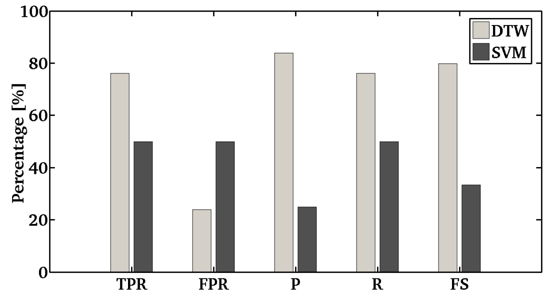
TABLE XIV: SRF and SRB accuracies comparison.

Percentages with SVM (table left) and with DTW (table right).

	SRF	**SRB**
SRF	100	0
SRB	100	0

	SRF	**SRB**
SRF	100	0
SRB	47.82	52.17

Fig. 2: Comparison between SVM and DTW for SR case.



IX. Conclusions

In this paper, two recognition tasks have been considered: Activity Recognition (AR) and Movement Recognition (MR). AR detects if a patient is idle, still, walking, running, going up or down the stairs or cycling while MR allows recognizing specific movements often required for physical rehabilitation such as arm circles, arm presses, arm twist, curls, seaweed and shoulder rolls. Smartphones are the reference platforms. The considered accelerometer signal classification techniques are: two kind of Support Vector Machine (SVM) known as *One-Against-All* (OAA) and *One-Against-One* (OAO); a Decision Tree (DT) based on the C4.5 training algorithm; the Dynamic Time Warping (DTW) that, differently from the previously listed techniques, does not employ signal features but directly operates on the signal's samples. The obtained results allow concluding that AR and MR functions can be successfully implemented over smartphones. In particular, the *SVM-OAO* exhibits the best accuracy in the AR case (91%) and in the MR case (99%). A further interesting conclusion, regarding MR, concerns the cases in which could be necessary distinguish among *ArmCircleFront* and *ArmCircleBack* or *ShoulderRolls-Front* and *ShoulderRollsBack*: SVM classifiers are not able to provide good results while DTW performs better. It means that fine classifications of the considered movements may need joint implementations of the two methods.

References

[1] H. C. Noel, D. C. Vogel, J. J. Erdos, D. Cornwall, and F. Levin, "Home telehealth reduces healthcare costs," *Telemedicine Journal & E-Health*, vol. 10, no. 2, pp. 170–183, 2004.

[2] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous wsn for healthcare: Recent advances and future prospects," *IEEE Internet of Things Journal*, vol. 1, no. 4, 2014.

[3] R. C. A. Alves, L. B. Gabriel, B. T. de Oliveira, C. B. Margi, and F. C. L. dos Santos, "Assisting physical (hydro)therapy with wireless sensors networks," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 113–120, April 2015.

[4] B. Mortazavi, M. Pourhomayoun, H. Ghasemzadeh, R. Jafari, C. K. Roberts, and M. Sarrafzadeh, "Context-aware data processing to enhance quality of measurements in wireless health systems: An application to met calculation of exergaming actions," *IEEE Internet of Things Journal*, vol. 2, no. 1, pp. 84–93, Feb 2015.

[5] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "A smartphone-centric platform for remote health monitoring of heart failure," *International Journal of Communication Systems*, vol. 28, no. 11, pp. 1753–1771, 2015. [Online]. Available: http://dx.doi.org/10.1002/dac.2778

[6] ——, "Smartphone-centric ambient assisted living platform for patients suffering from co-morbidities monitoring," *Communications Magazine, IEEE*, vol. 53, no. 1, pp. 34–41, January 2015.

[7] C. Garritty and K. El Emam, "Whos using pdas? estimates of pda use by health care providers: a systematic review of surveys," *Journal of medical Internet research*, vol. 8, no. 2, 2006.

[8] B. G. Steele, B. Belza, J. Hunziker, L. Holt, M. Legro, J. Coppersmith, D. Buchner, and S. Lakshminaryan, "Monitoring daily activity during pulmonary rehabilitation using a triaxial accelerometer," *Journal of Cardiopulmonary Rehabilitation and Prevention*, vol. 23, no. 2, pp. 139–142, 2003.

[9] A. Marshall, O. Medvedev, and A. Antonov, "Use of a smartphone for improved self-management of pulmonary rehabilitation," *International journal of telemedicine and applications*, vol. 2008, p. 2, 2008.

[10] M. Varnfield, M. Karunanithi, C.-K. Lee, E. Honeyman, D. Arnold, H. Ding, C. Smith, and D. L. Walters, "Smartphone-based home care model improved use of cardiac rehabilitation in postmyocardial infarction patients: results from a randomised controlled trial," *Heart*, pp. heartjnl–2014, 2014.

[11] A. Sarela, J. Salminen, E. Koskinen, O. Kirkeby, I. Korhonen, and D. Walters, "A home-based care model for outpatient cardiac rehabilitation based on mobile technologies," in *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*. IEEE, 2009, pp. 1–8.

[12] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive computing*. Springer, 2004, pp. 1–17.

[13] N. Győrbíró, Á. Fábián, and G. Hományi, "An activity recognition system for mobile phones," *Mobile Networks and Applications*, vol. 14, no. 1, pp. 82–91, 2009.

[14] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 156–167, 2006.

[15] J. Fahrenberg, F. Foerster, M. Smeja, and W. MÜLLER, "Assessment of posture and motion by multichannel piezoresistive accelerometer recordings," *Psychophysiology*, vol. 34, no. 5, pp. 607–612, 1997.

[16] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *Ubiquitous Computing Systems*. Springer, 2006, pp. 516–527.

[17] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 1, pp. 20–26, 2008.

[18] A. Sugimoto, Y. Hara, T. Findley, and K. Yoncmoto, "A useful method for measuring daily physical activity by a three-direction monitor." *Scandinavian Journal of Rehabilitation Medicine*, vol. 29, no. 1, pp. 37–42, 1997.

[19] M. Sekine, T. Tamura, M. Akay, T. Fujimoto, T. Togawa, and Y. Fukui, "Discrimination of walking patterns using wavelet-based fractal analysis," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 10, no. 3, pp. 188–196, 2002.

[20] T. Tamura, M. Sekine, M. Ogawa, T. Togawa, and Y. Fukui, "Classification of acceleration waveforms during walking by wavelet transform." *Methods of information in medicine*, vol. 36, no. 4-5, pp. 356–359, 1997.

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[22] S.-k. Song, J. Jang, and S. Park, "A phone for human activity recognition using triaxial acceleration sensor," in *International Conference on Consumer Electronics*. IEEE, 2008, pp. 1–2.

[23] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*. IEEE, 2010, pp. 1–7.

[24] D. Maguire and R. Frisby, "Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data," in *9th. IT & T Conference*, 2009, p. 11.

[25] J.-Y. Yang, Y.-P. Chen, G.-Y. Lee, S.-N. Liou, and J.-S. Wang, "Activity recognition using one triaxial accelerometer: A neuro-fuzzy classifier with feature reduction," in *Entertainment Computing–ICEC 2007*. Springer, 2007, pp. 395–400.

[26] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection," in *Wireless sensor networks*. Springer, 2008, pp. 17–33.

[27] R. K. Ganti, P. Jayachandran, T. F. Abdelzaher, and J. A. Stankovic, "Satire: a software architecture for smart attire," in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 110–123.

[28] S. L. Lau and K. David, "Movement recognition using the accelerometer in smartphones," in *Future Network and Mobile Summit, 2010*, June 2010, pp. 1–9.

[29] J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, no. 5, 2006.

[30] Z. Prekopcsák, P. Halácsy, and C. Gáspár-Papanek, "Design and development of an everyday hand gesture interface," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 2008, pp. 479–480.

[31] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[32] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, Feb 1978.

[33] X. Hui and H. Yunfa, "Data mining based on segmented time warping distance in time series database [j]," *Journal of computer research and development*, vol. 1, p. 010, 2005.

[34] P. H. Eilers, "Parametric time warping," *Analytical chemistry*, vol. 76, no. 2, pp. 404–411, 2004.

[35] J. A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study," *The Journal of the Acoustical Society of America*, vol. 103, no. 4, pp. 2185–2196, 1998.

[36] J. Blackburn and E. Ribeiro, "Human motion recognition using isomap and dynamic time warping," in *Human Motion–Understanding, Modeling, Capture and Animation*. Springer, 2007, pp. 285–298.

[37] A. Corradini, "Dynamic time warping for off-line recognition of a small gesture vocabulary," in *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, 2001, pp. 82–89.

[38] DoctorOz. (2011) 5-minute arm workout. [Online]. Available: www.youtube.com/watch?v=1JCGybW-zaw

[39] M.-K. Suh, C.-A. Chen, J. Woodbridge, M. K. Tu, J. I. Kim, A. Nahapetian, L. S. Evangelista, and M. Sarrafzadeh, "A remote patient monitoring system for congestive heart failure," *J. Med. Syst.*, vol. 35, no. 5, pp. 1165–1179, Oct. 2011.

[40] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, Mar 2002.

[41] M. Fahim, I. Fatima, S. Lee, and Y.-K. Lee, "Activity recognition based on svm kernel fusion in smart home," vol. 203, pp. 283–290, 2012.

[42] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[43] ——. LIBSVM: A library for support vector machines. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[44] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, ser. CIKM '00. New York, NY, USA: ACM, 2000, pp. 20–29. [Online]. Available: http://doi.acm.org/10.1145/354756.354775

[45] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[46] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[47] Digital Signal Processing Laboratory, University of Genoa, Italy. (2016) Accelerometer signal set for AR and MR with smartphones. [Online]. Available: http://www.dsp.diten.unige.it/resources/accDB

[48] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427 – 437, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457309000259