Pers Ubiquit Comput (2013) 17:771–782
DOI 10.1007/s00779-012-0626-y

ORIGINAL ARTICLE

Exercise repetition detection for resistance training based on smartphones

Igor Pernek · Karin Anna Hummel · Peter Kokol

Received: 1 May 2012 / Accepted: 7 November 2012 / Published online: 6 December 2012

Abstract Regular exercise is one of the most important factors in maintaining a good state of health. In the past, different systems have been proposed to assist people when exercising. While most of those systems focus only on cardio exercises such as running and cycling, we exploit smartphones to support leisure activities with a focus on resistance training. We describe how off-the-shelf smartphones without additional external sensors can be leveraged to capture resistance training data and to give reliable training feedback. We introduce a dynamic time warping-based algorithm to detect individual resistance training repetitions from the smartphone's acceleration stream. We evaluate the algorithm in terms of the number of correctly recognized repetitions. Additionally, for providing feedback about the quality of repetitions, we use the duration of an individual repetition and analyze how accurately start and end times of repetitions can be detected by our algorithm. Our evaluations are based on 3,598 repetitions performed by ten volunteers exercising in two distinct scenarios, a gym and a natural environment. The results show an overall repetition miscount rate of about 1 % and overall temporal detection error of about 11 % of individual repetition duration.

Keywords Wearable systems · Resistance training · Smartphone · Accelerometer

1 Introduction

Physical activity is one of the most important factors in maintaining a good state of health. Regular and proper exercising reduces many chronic diseases such as cardiovascular disease, diabetes, cancer, hypertension, obesity, depression, and osteoporosis [8, 17, 28, 36] and can even outperform medical treatment in specific cases [25]. The benefits of physical activity can therefore also be measured in economic sense, especially in terms of reduced health care costs.

Recently, a multitude of both commercial products and research prototypes have emerged that support people during exercising. Different vendors such as Garmin, Polar, and Suunto, developed GPS enabled wristwatch computers. They are able to capture training data (e.g., distance, duration, and training pace) and upload it to a personal computer or to a dedicated Internet-based service. Such services support long-term data collection and allow users to track their progress over time and share training results with their peers.

With smartphones being ubiquitous and powerful portable computers, smartphone-based training assistants are gaining a lot of attention. The main reasons behind this are their ability to capture data from different sensors, analyze it in real-time, provide immediate training performance feedback, and even store captured data for further analysis and inspection [2, 9]. Modern application delivery

I. Pernek (✉) · P. Kokol
Faculty of Health Sciences, Research Institute,
University of Maribor, Zitna ulica 15,
2000 Maribor, Slovenia
e-mail: igor.pernek@uni-mb.si

P. Kokol
e-mail: kokol@uni-mb.si

K. A. Hummel
Computer Engineering and Networks Laboratory,
Communication Systems Group, ETH Zürich,
Gloriastrasse 35, 8092 Zurich, Switzerland
e-mail: karin.hummel@tik.ee.ethz.ch

channels, such as Apple's App Store and Google's Play (formerly, Android's Marketplace), are easing the distribution and installation of third-party training applications, thus, transforming off-the-shelf smartphones into powerful training assistants. However, most of the available training assistants track only cardio training such as cycling, running, hiking, etc.

Although studies have shown that in addition to cardio training, resistance training is an important part of a balanced exercise program [3, 17], there is still a void in the area of automated resistance training assistants. Thus, resistance training diaries are often still paper-based or rely on manual input to smartphones such as Jefit [18]. As manual input is cumbersome, taking notes about resistance training is often avoided, which can negatively affect people's activity levels and decrease their motivation for regular training [32, 34]. While there are some attempts to automate resistance training tracking, these solutions are either not ubiquitous, targeted to a set of specific exercises, or they do not allow fine-grained tracking of individual repetitions. The latter is important to give feedback about the correctness and quality of exercising.

In this paper, we leverage off-the-shelf smartphones as a ubiquitous platform for capturing resistance training information to reach a broad range of people. As most of the standards, guidelines, and position statements regarding physical activity prescribe resistance training in terms of sets and repetitions [17], we investigate how smartphones can automatically capture the number of repetitions and provide feedback on the performance of individual repetitions. To demonstrate the feasibility, we create a resistance training assistant, which can be run on any accelerometer equipped Android phone. We choose to rely on accelerometer sensors as they have already been validated for measuring resistance training performance [30] and are becoming an integral part of most smartphones [1]. The assistant guides users through a preset resistance training program and captures acceleration data for different exercises. We introduce an algorithm based on dynamic time warping (DTW) to process the acquired acceleration stream in real-time and derive information such as the number of repetitions and the duration of individual repetitions. Capturing the duration of repetitions is important because duration can be used as one of the metrics for providing qualitative feedback. Resistance training literature suggests that long repetition durations are effective for building strength, and that short repetition durations are beneficial for speed and power gain [14]. Consequently, correctly recognized repetition duration can be used to give accurate feedback to users about the quality of exercising by rating the duration spent on a particular exercise.

Considering the fact that resistance training can be performed in different environments such as in the gym, at home, and outdoors, and while using different types of equipment such as weight-stack machines, free weights, and resistance bands, we demand the algorithm to be robust and general in order to perform in different settings. During evaluation, we challenge the algorithm with a set of commonly used exercises in two different environments, the constrained gym and the unconstrained outdoors environment.

The contributions of this paper are summarized as follows:

- We enable off-the-shelf smartphones to capture resistance training information for different exercises and types of equipment based on incorporated acceleration sensors. To the best of our knowledge, no existing commercial or academic work exploits smartphones for tracking such a broad range of different resistance training exercises.
- We design a robust dynamic time warping-based algorithm suitable for smartphones to detect exercise repetitions in a continuous acceleration stream. The algorithm is able to detect start and end times of repetitions. Consequently, the number and duration of repetitions can be tracked.
- We present an experimental study to evaluate the algorithm's accuracy in terms of correctly detected repetitions and temporal accuracy of detection of repetition start and end times. Therefore, we conducted experiments observing ten users in two different environments while performing nine different exercises. We describe the outcomes and summarize the results.

The rest of the paper is organized as follows: After presenting related work in Sect. 2, we describe how smartphones can be used to provide resistance training assistance and introduce our Android prototype application in Sect. 3. Section 4 introduces the repetition detection algorithm. Finally, we present the results of the experimental evaluation of the algorithm in Sect. 5 and conclude the paper in Sect. 6.

2 Related work

In the past, only a few attempts have been made towards automated resistance training data collection. Fitlinxx [13] proposes a solution where computer systems with touch screens are mounted to weight machines in gyms. Users have to log in to the system prior to performing any exercise, and the machine automatically stores the number of repetitions the users perform into their profile. A mechanism hooked to the machine's weight-lifting transmission system captures the exercise signal and extracts the number of repetitions. A study [4] has shown that the Fitlinxx system indeed can motivate users to do more resistance training. However, the system is very expensive, as the gym has to be equipped with specific exercise machines and infrastructure, and has therefore not gained broader popularity.

Mattman et al. [20] propose a garment prototype using strain sensors to recognize upper body postures. They evaluate the system for a set of gym exercises and show that the system is able to support resistance training exercises. It is shown that textile worn strain sensors can be used to infer the speed of movement, the repetition frequency, and the number of resistance training repetitions. However, no algorithm is provided that can be directly applied to acceleration sensors.

Meltzi et al. [21] propose a wireless body area network system for supervision of resistance training exercises. Accelerometer equipped bands are used to capture the movement stream for a simple resistance training exercise. Further, a PC is used to analyze the data and to provide feedback on the quality of exercising. However, the algorithm used for analyzing the data relies on exercise specific features, such as the position of the elbow during the biceps curl exercise, and can therefore not be easily applied to a general resistance training exercise.

Asselin et al. [6] describe a wearable solution that supports monitoring of a range of cardio and resistance training exercises. To track resistance training, the system uses wearable accelerometers. The acceleration stream is transferred to a laptop computer and processed by using a threshold-based low-pass filtering algorithm. The signal is compared to a dynamically set threshold value to distinguish exercising and resting state. Hence, the system observes each transition from resting to exercise state to count the number of repetitions. Further, the number of time units in each state is analyzed to detect struggling during exercise execution. However, this approach assumes that all the repetitions have a similar single peak acceleration footprint, which is not true for exercises performed in different environments.

The system proposed in [10] offers capabilities for tracking free-weight exercises. It is based on a mobile phone gateway connected to two external accelerometer sensors. The system is able to recognize exercises performed with free weights and count the number of exercise repetitions. Due to the design of the acceleration capturing mechanism using data from a glove and a chest belt, the system is only able to derive information for a limited set of free-weight exercises. In addition to the limited usability, the number of repetitions is counted based on simple peak detection algorithms that are not able to detect a repetition's start and end time and, therefore, lack capabilities for providing qualitative feedback based on the duration of repetitions. Seeger et al. [33] advance this solution by making the repetition detection algorithm robust to different exercising speeds.

Muehlbauer et al. [22] propose an approach that is the most similar to our work. This solution exploits arm worn smartphones to recognize a number of upper body resistance training exercises from a continuous acceleration stream. While the main focus of this work lies on spotting series of exercises, it also contains an algorithm for detecting the number of repetitions. Similar to the work described in [6, 10], repetitions are counted based on peak detection. Consequently, start and end times of individual repetitions cannot be detected. We advance this solution by supporting a broader set of exercises, which can be performed using different types of equipment, and by detection of duration of individual repetitions.

To summarize, the discussed related systems are either not ubiquitous or use very basic repetition detection algorithms often limited to specific exercises and dependent on dedicated sensors. As the proposed algorithms are mostly based on counting the peaks in the signal, they are not able to detect start and end times of repetitions and consequently lack the capability for assessing quality of repetitions based on their duration. We address these limitations and present a robust algorithm suitable for smartphones that is able to detect individual repetitions and their duration.

3 Resistance training assistant

To become ubiquitously available, the resistance training assistant has to fulfill a set of requirements. First, the used platform should not add high extra cost nor demand the cumbersome installation of additional sensors and tools. Second, the assistant should provide an easy to use interface, be encouraging, for example, by suggesting particular exercises, and give feedback on the exercising performance as well as the training progress. Finally, the prototype should be usable in different environments including gyms and natural environments and cover a wide range of exercises.

3.1 Platform

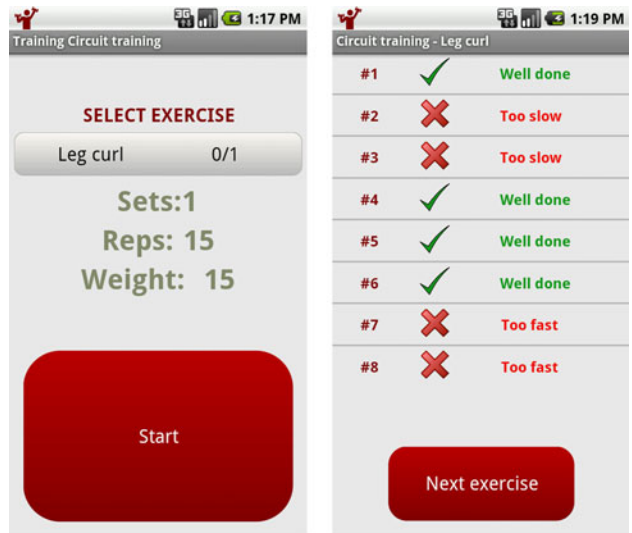
We target our research on providing a resistance training solution utilizing solely off-the-shelf smartphones. While previous approaches [6, 10, 21] use additional external sensors, we only use onboard sensors. In our analysis, we will evaluate whether our approach is able to provide sufficiently accurate acceleration readings to detect exercising reliably. Hence, we state a minimum requirement for smartphones, that is, the presence of an accelerometer sensor, as we exploit the captured acceleration signal for automated extrapolation of resistance training information. Although being lightweight and portable, smartphones offer powerful processing capabilities, sufficient memory, large and intuitive to operate (touch) screens, and advanced wireless communication options. With built-in microphones and speakers, they are also able to provide audio feedback and accept voice commands. Therefore, smartphones are a very convenient tool for creating sport assistants that are inexpensive, accessible to a wide range of people, easy to carry around, and usable in different environments.

As a consequence, the mobile application prototype is developed to run on accelerometer equipped smartphones with at least Android 2.1 installed (as operating system). From a software development perspective, the prototype implementation consists of five “activities,” seven “views,” and two “services” and is structured into the following main components: (1) user interface, (2) sensor sampling service, (3) settings, (4) repetition recognition, and (5) network communication. The application allows users to download different training plans and exercises from an online training database. After selecting a training plan, the application proposes exercises to perform along with some additional information such as the number of repetitions required or the intensity, that is, weight to be aimed for (see Fig. 1a). Further, the assistant provides real-time feedback about the desired exercise effect by evaluating the duration of the repetitions (Fig. 1b). Additionally, the application permanently stores the number and the intensity of repetitions for each exercise and allows users to upload it to an online profile to track progress and observe exercising trends (if wanted, this information may be shared within an online community).

3.2 Exercise environments

Resistance training can be performed in different environments using different types of exercising equipment. To provide a solution useful not only to a small group of people training in a specific environment, we support a range of commonly recommended exercises targeting the major muscle groups and exemplify the use of the training assistant in two environments: the gym and a non-dedicated or outdoor environment.

Most common resistance training exercises are performed with the help of exercise machines, free weights, or resistance bands. Exercise machines are usually large, heavy, and immobile and are therefore mostly limited to indoor environments (e.g., gyms). On the other side, free weights and resistance bands are portable and can also be used outdoors. Supporting both environments is crucial, as they both have advantages that make them appealing. For example, gym training is not affected by weather conditions and can therefore provide 24/7 availability of training options during cold or rainy periods. Furthermore, exercise machines, which are the mainstream type of equipment in many gyms, can easily be used by people without any resistance training experience as they constrain the path of the movement during the exercise to a fixed trajectory. This characteristic is also exploited to ease the automatic detection of exercise repetitions. However, a lot of people prefer outdoor activities. Thus, free weight and resistance bands are a lightweight and low-cost alternative. Here, the training assistant has to cope with movement not as constrained as in the gym environment.



(a) (b)

Fig. 1 Screenshots of the mobile application showing (**a**) training information [suggested number of sets, repetitions, and suggested weight to be used (in kg)] for one sample exercise and (**b**) feedback on recognized repetitions

To support both training environments in our approach, we cover resistance training using either exercise machines or resistance bands and/or free weights. Additionally, all of the free weight and resistance band exercises can also be performed without any equipment using solely body weight of the person exercising.

Exercises performed on exercise machines are termed *constrained exercises* referring to the constrained path of smartphone movement during exercise execution. For exercise machine training, the smartphone is positioned on the stack of weights and, thus, is constrained to vertical movement only (Fig. 2b). Figure 2c depicts an alternative placement of the smartphone on an exercise machine. Exercises performed with free weights, resistance bands, or body weight are termed *unconstrained exercises* referring to the increased freedom in the movement path of the smartphone during training. The last column of Table 1

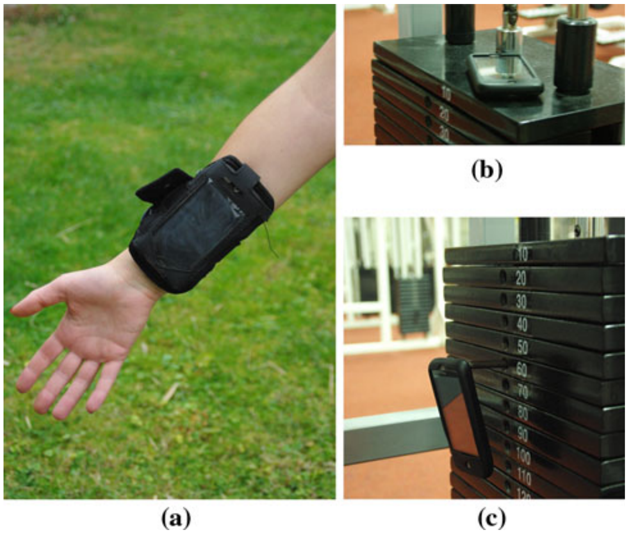


Fig. 2 Different placements of the smartphone: (**a**) wrist placement in the unconstrained training environment, (**b**), (**c**) two alternative placements in the constrained training environment (exercise machine)

provides the smartphone position for different unconstrained exercises. Figure 2a shows the wrist placement of the smartphone in the unconstrained environment.

3.3 Muscle groups and exercising

A set of resistance training exercises included different muscles into constrained and unconstrained training. In [26], it is suggested that resistance training programs should contain a minimum of one exercise per each major muscle group. Based on recommendations and insights from resistance training literature [7, 11, 24], we select exercises for all the major muscle groups for studying the performance of training in both the constrained and the unconstrained environment. The resulting exercise categories and number of exercises per category are as follows: four exercises for legs, two exercises for arms, two exercise for the upper body, and one exercise for the lower body. Table 1 provides a complete lists of the selected representative exercises that are included in our experimental study.

The detailed body movement depends on the equipment used for exercising. In the constrained environment, all of the exercises are performed using standard exercise machines. Differently, unconstrained exercises can be performed using multiple "types of equipment" such as free weights, resistance bands, or body weight only. Our preliminary tests showed that the output acceleration signal is similar for the different types of equipment used. In our experimental study, mainly resistance bands are selected as they are convenient to bring along. In case of sit-ups, squats, and calf raise exercises, however, resistance bands are not comfortable, and free weights or body weight of the person exercising is chosen.

4 Repetition detection algorithm

We propose an algorithm for accurate detection of resistance training repetitions in the acceleration signal. We do not offer solutions for detecting the type of the exercises performed as this has already been thoroughly researched by related work [10, 22]. Prior to describing the algorithm itself, we point the reader to Table 2, which summarizes the notation used in the remainder of the paper. For reasons of simplicity, repetition start and end times are given as indices of the start and end of the repetition in the acceleration stream.

A resistance training repetition is characterized by a specific acceleration pattern observed by the smartphone accelerometer sensor. Once recognized, *repetitions series* of these patterns can be observed. The patterns are a part of the continuous acceleration stream time series R. The goal

Table 1 List of constrained and unconstrained movement exercises for different muscles and muscle groups and placement of the smartphone in the unconstrained environment (in the constrained environment, the smartphone is placed on the top of the weights)

Muscle group	Muscle	Constrained exercise	Unconstrained exercise	Placement unconstrained
Legs	Gluteus maximus	Leg press	Squats	Ankle
Legs	Hamstrings	Lying leg curl	Standing resistance band leg curl	Ankle
Legs	Quadriceps	Seated leg extension	Seated resistance band leg extension	Ankle
Legs	Soleus	Seated calf raises	Standing calf raises	Ankle
Arms	Triceps brachii	Machine seated push-downs	Resistance band triceps extensions	Wrist
Arms	Biceps brachii	Machine low-pulley curls	Resistance band biceps curls	Wrist
Upper body	Pectoralis major	Bench press	Resistance band crossover flys	Wrist
Upper body	Latissimus dorsi	Seated back lat pull-downs	Standing back lat pull-downs	Wrist
Lower body	Rectus abdominis	Machine crunches	Sit-ups	Wrist

Table 2 A summary of algorithm notation

C	Repetition pattern time series $C = c_1, \ldots, c_n$
n	Number of samples in repetition pattern time series C
R	Acceleration stream time series $R = r_1, \ldots, r_m$
p	Peak index in repetition pattern time series, calculated as $max(C)$
I	Indices of repetition candidates (peaks in acceleration stream R) $I = i_1, \ldots, i_o$
ϵ	Expansion value for varying the window around repetition candidates
α	Peak similarity factor
s	Approximate repetition start time
e	Approximate repetition end time
s'	Exact repetition start time
e'	Exact repetition end time

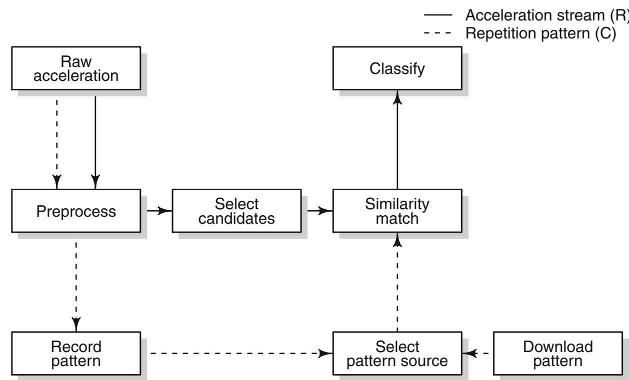


Fig. 3 Overview of the DTW-based repetition detection algorithm. The processing steps for generating the repetition pattern C are indicated with *dashed arrows,* while *solid arrows* depict the processing path for the acceleration stream R

of the repetition detection algorithm is to accurately detect the acceleration pattern corresponding to the movement on a resource constrained smartphone in real-time. For detection, repetition patterns (C) are used, which could either be prerecorded by a resistance training expert and downloaded from the Internet or calibrated once for each individual user (e.g., during the first training session under the supervision of a personal trainer). During resistance training, each pattern C is matched against subsections of the acceleration stream R and compared in terms of similarity. However, repetition pattern C and an actual repetition contained in R are rarely perfectly temporally aligned. A distance metric such as Euclidean distance is not able to capture these time shifts and is of little use for any similarity comparison. We therefore propose the use of dynamic time warping (DTW) [29] for measuring the similarity between an actual repetition and a repetition pattern C. DTW is a dynamic programming technique, which has already been used for activity [23] and gesture recognition [19]. DTW allows to compare sequences which are not temporally aligned by producing a mapping minimizing the distance between the input sequences. Furthermore, the mapping produced can be leveraged to gain information about the location of specific events in the acceleration stream R that where previously annotated in a repetition pattern C such as the repetition start and end time. In [35], DTW matching for data streams has been proposed similar to our setting. However, as DTW processing is computationally too intensive to be performed continuously on a smartphone, we introduce additional steps to improve the efficiency of the algorithm.

Figure 3 depicts the main steps of our approach. First, a *preprocessing* step is applied to *raw acceleration* signals (to generate R and C). The pattern signal C is stored in the pattern repository in the step *record pattern*. In case of a standalone personal trainer scenario without self-calibration, the pattern C can also be downloaded from online sources (step *download pattern*). During exercising, R is preprocessed and skimmed for a list of promising repetition candidates in the step *select candidates*. As a consequence, DTW calculation, that is, *similarity matching* of C against a subset of the original R, is performed only for a reduced set of candidates. This makes the algorithm faster and a lot less computationally intensive. Finally, the results from similarity matching are passed to the *classification* step resulting in a set of detected repetitions. We detail each step of our approach in the following sections.

4.1 Preprocessing

Preprocessing is applied to both the repetition pattern C and acceleration stream R and consists of the following operations: uniform resampling, signal smoothing, and detecting the major axis.

As accelerometers incorporated into smartphones were primarily designed for occasionally adapting the user interface based on the device orientation and not for continuous sensing, they do not maintain a constant sampling rate all of the time. Therefore, to be able to assure optimal DTW matching results, the acceleration signal first needs to be uniformly resampled. To achieve this, we apply linear interpolation when the sample frequency is too low and averaging of consecutive points when the samples are too dense. For the purpose of evaluation, the output sampling frequency was set to 10 Hz (one sample every 100 ms). As resistance training repetitions are usually performed with much lower frequency, sampling at 10 Hz provides a good compromise between signal compression effort (and consequently less processing) and quality.

Additionally, the signal acquired is smoothed with a seven point frame third order Savitzky-Golay (SG) smoothing filter [31]. The filter values have been selected based on the performance of the algorithm in an experimental pre-study of different resistance training signals (note that the pre-study is based on different data than the experimentation study presented in Sect. 5 for evaluation). The SG filter approximates the values within a specified window by a polynomial of a specified order minimizing its least-square error. The advantage of using SG filtering is that it does not delay the signal and is able to preserve features such as local minima and maxima.

To reduce computational effort while doing similarity matching, the *major axis* is detected for each exercise. Although smartphones are capable of sensing acceleration in three different directions, the exercises impose the majority of acceleration almost always in one direction only. We call this direction the major axis and detect it in the last preprocessing step of the algorithm for each exercise. For simplicity reasons, the selection of the major axis is performed only once for each exercise during the calibration process. Therefore, smartphone positioning during repetition detection has to match the calibration positioning. However, this constraint could easily be relaxed by repeating the process of major axis detection at the beginning of each exercise.

4.2 Selection of candidates

Continuous DTW calculation in a streaming environment is too time-consuming to be performed in real-time on a smartphone. Thus, to decrease the time-complexity of the algorithm, we propose an additional step that eliminates the need to calculate the DTW distance for subsequences of R that are obviously not exercise repetitions.

The intuition behind our approach is that most of the exercise repetitions result in acceleration peaks of similar magnitude. Therefore, a threshold-based derivative peak detection algorithm similar to the one proposed in [5] is used to locate peaks with intensity similar to $max(C)$ for a given exercise. Peak indices found are passed to the similarity matching phase, where a pattern with the shortest normalized DTW distance is found in the neighborhood of each peak.

A threshold is used to omit peaks with amplitude much lower than the maximum of acceleration in C. In detail, only peaks i where $R_{[i:i]} \geq max(C) \cdot \alpha^{-1}$ are passed to the similarity matching phase. The parameter α denotes the level of peak similarity demanded by the algorithm. The higher the value of α the more peaks are passed to the similarity matching phase, and the smaller the number of false negatives is. At the same time, the number of repetition candidates grows, and the DTW distance has to be calculated more frequently, which increases the computational cost. Therefore, a reasonable value for α has to be chosen for optimal performance. In our implementation, we selected $\alpha = 3$, which results in omitting all peaks with a value lower than one third of the maximum value of C. The selection of α is based on the observation that in our pre-study, most of the peaks that occurred due to noise or short non-repetition movements before and after each exercise had an amplitude lower than one third of $max(C)$ and are therefore efficiently filtered.

4.3 Similarity matching

Algorithm 1 outlines the DTW similarity matching step for a pattern C (Fig. 4 visualizes the concept). The algorithm loops through the repetition candidates' indices in I obtained from the previous phase of the algorithm and calculates the approximate repetition start time s and end time e, both in terms of indices (lines 3–4, Algorithm 1). These times are later used to extract subsequences of R for DTW calculation. Information inhibited in the calibration move structure, that is, the number of samples n and the peak index p, is used to infer s and e.

The basic calculation considers the repetition's pattern C peak index $p = max(C)$, which is mapped to the peak index i of the movement pattern in the acceleration stream R. The start time is calculated as $s = i - p$, and the end time is calculated as $e = i + (n - p)$. For example, given a repetition pattern C that contains $n = 30$ samples and a maximum value (peak) at index $p = 10$, the approximate start index s and end index e for a repetition candidate with maximum value at index $i = 50$ will be: $s = 40$ and $e = 70$.

However, as the duration of an actual repetition can be longer than the duration of the recorded repetition pattern C, we use an expansion value ϵ to introduce a configurable

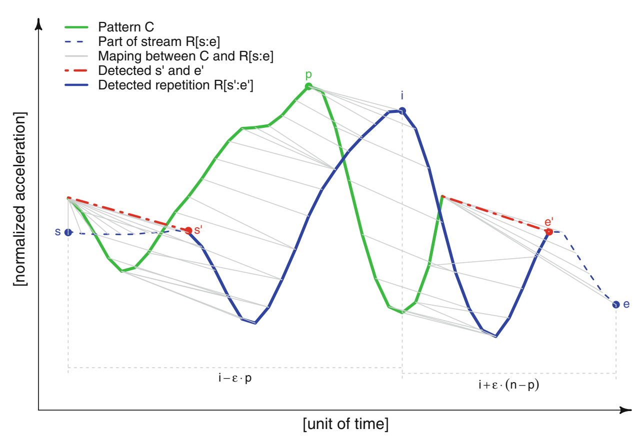


Fig. 4 DTW mapping between repetition pattern C and a part of acceleration stream R along with annotations of important events (Table 2 summarizes the notation used)

tolerance offset. An expansion value of $\epsilon = 1$ means that the subsequence $R_{[s:e]}$ is of the same length as the repetition pattern C. In our implementation the expansion value $\epsilon = 1.5$ is used, which means the subsequences extracted from R have an extra 50 % margin added on each side. Using such value for ϵ allows us to capture repetitions that are up to 100 % longer than C, which has been the upper bound of the difference between repetition patterns and actual repetitions' length in our pre-study.

Unconstrained DTW matching as described in [35] is used to find a subsequence of $R_{[s:e]}$ that is the most similar to the repetition pattern C and to produce the DTW mapping (*map*) between matched signals (Algorithm 1, line 5). The mapping is used to infer the positions of exact repetition start index s' and end index e' (Algorithm 1, line 6–7) as depicted by the dashed red line in Fig. 4. The use of an unconstrained DTW algorithm allows us to use an approximate sequence $R_{[s:e]}$ to find an exact repetition candidate $R_{[s':e']}$.

A set of features is calculated for each subsequence $R_{[s':e']}$ (Algorithm 1, line 8–14), they are: normalized DTW distance (*dst*), maximum (*max*), minimum (*min*), arithmetic mean (*mean*), standard deviation (*sd*), root mean square (*rms*), and duration (*dur*). These features have very low computational costs or are already precalculated in previous steps of the algorithm (normalized DTW distance). Instruction on how these features are calculated can be found in [12]. Finally, s' and e', and the features calculated are passed to the classification step.

Algorithm 1 $DTWmatch(C, R, p, I, \epsilon)$

1: $\Re \Leftarrow \emptyset$
2: **for all** i in I **do**
3: $s \Leftarrow i - \epsilon \cdot p$
4: $e \Leftarrow i + \epsilon \cdot (n - p)$
5: $map \Leftarrow calculateDTW(C_{[1:n]}, R_{[s:e]})$
6: $s' \Leftarrow$ last element in R translating into $C_{[1]}$ in map
7: $e' \Leftarrow$ first element in R translating into $C_{[n]}$ in map
8: $dst \Leftarrow normalizedDTWDistance(map, s', e')$
9: $max \Leftarrow maximum(R_{[s':e']})$
10: $min \Leftarrow minimum(R_{[s':e']})$
11: $mean \Leftarrow arithmeticMean(R_{[s':e']})$
12: $sd \Leftarrow standardDeviation(R_{[s':e']})$
13: $rms \Leftarrow rootMeanSquare(R_{[s':e']})$
14: $dur \Leftarrow duration(s' : e')$
15: $\Re \Leftarrow add(s', e', dst, max, min, mean, sd, rms, dur)$
16: **end for**

4.4 Classification

In the final step of the algorithm, each repetition pattern bounded by the tuple (s', e') is evaluated by a classifier to filter out probable false positives. Due to low computational demands and the observation that true and false positives detected during similarity matching are linearly separable, a logistic regression classifier is applied.

Logistic regression is a classification method that uses one or more independent variables to predict a binary dependent variable. Training a logistic regression classifier generates a linear formula (Eq. 1) with coefficients b_i. The formula is used to calculate a transformation of an acceleration pattern to finally evaluate, whether this is a valid repetition.

$$\begin{aligned} logit(p) = b_0 + b_1 \cdot dst + b_2 \cdot max + b_3 \cdot min \\ + b_4 \cdot sd + b_5 \cdot rms + b_6 \cdot dur \end{aligned} \tag{1}$$

The logistic regression coefficients are trained with the help of the Weka machine learning toolbox [16]. To prevent over fitting, ten-fold cross-validation is used. As different exercises in different environments result in different acceleration patterns, 18 different models for nine exercises and two exercising environments are created. Because each model is defined by only seven numerical coefficients, this solution does not introduce a lot of computation and storage overhead.

In the classification step, model parameters are selected with respect to the current exercising environment and exercise performed. The parameters are fed into Eq. 1 to calculate the $logit(p)$ value. Finally, the $logit(p)$ value of each acceleration pattern is transformed to a probability value and evaluated by the rule given by Eq. 2 to accept only patterns that are more probable to be actual repetitions.

$$Eval(s', e') = \begin{cases} \text{accept}, & \text{if } \frac{1}{1+e^{-logit(p)}} \geq 0.5 \\ \text{reject}, & \text{if } \frac{1}{1+e^{-logit(p)}} < 0.5 \end{cases} \tag{2}$$

5 Evaluation

To give an answer to the question whether the proposed repetition detection algorithm is capable to detect exercise repetitions accurately using off-the-shelf smartphone accelerometers, we conducted experiments in the constrained (weight machines) and unconstrained environment (resistance bands, free weights, and body weight). In this section, we present details about the experiment, data collection, and the results of the evaluation in terms of accuracy of repetition detection and accuracy of detecting the start and end times of repetitions.

5.1 Data collection

Accelerometer data were collected using the HTC Desire smartphone with a built-in 3-axis accelerometer (Bosch Sensortec BMA150). The captured acceleration signal consists of 708,606 samples for each of the 3 accelerometer axis. The actual start and end times of each repetition were marked through voice annotations. Voice annotations were not performed by the person exercising, but by an experienced resistance training practitioner, who was also supervising the correctness of the repetitions being performed. The ground truth established through voice annotations was used to assess the repetition detection algorithm in terms of the number of correctly recognized repetitions and repetition start and end time detection error. Prior to each series of repetitions, subjects were also asked to perform a calibration repetition that was stored in the repetition repository as repetition pattern C. We used audio notifications played by the smartphone to guide the subjects through the calibration process and to obtain calibration patterns with clearly defined start and end times. In total, 360 calibration moves and 3,598 exercise repetitions were collected[1].

A total of ten healthy subjects (six males, four females; age $= 25.6 \pm 10.8$ years) were recruited to perform the exercises listed in Table 1. All of the constrained environment exercises were performed in a gym on exercise machines. Unconstrained exercises were not tied to a specific environment; hence, they were performed in different places such as the subjects' homes, outdoors in the park, and some in the gym, but not using exercise machines. Each user was asked to perform two series of ten repetitions for each exercise in both environments.

As resistance training exercises can be performed with different weight loads, the load used for the first series was slightly lower as the one during the second series. Two approaches were used to specify the exercising load. In the constrained environment, one repetition maximum (1 RM) was measured for each user and exercise pair. One repetition maximum is the maximum amount of weight a person can lift in a single repetition. As loads greater than 50 % of the 1 RM have been shown to increase muscular strength [17], we have selected the load of 50 % of 1 RM for the first series and the load of 70 % of 1 RM for the second series. However, for unconstrained exercises, the relationship between different loads is not as precise as for constrained exercises, as, for example, resistance bands do not specify numerical weight load values. This means it is only possible to say that exercising with one resistance band, for example, a black band, is harder than with a green one, as the black resistance band is thicker than the green one. Therefore, we asked the subjects to experiment with resistance bands of different thickness and select the one they felt comfortable with when doing 15 repetitions for the first series and the one they felt comfortable with when doing 10 repetitions for the second one. Similar recommendations were given for using free weights and body weight.

5.2 Repetition detection accuracy

To assess the effectiveness of the repetition detection algorithm precision, recall, and F score were reported for each combination of user, exercise, and training environment. Precision, recall, and F score are standard information retrieval statistics that are commonly used for problems dealing with highly skewed datasets like ours, where the number of positive examples is much smaller than the number of negative ones. R [15,27] was used to calculate all the statistics.

Precision (Eq. 3), also known as positive prediction value, here relates the number of correctly recognized repetitions (true positives) to the number of all cases that resulted in a classification as a *repetition*.

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3}$$

Recall (Eq. 4), also known as specificity, is here the ratio of correctly recognized repetitions divided by all repetitions taking place that should have been recognized.

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{4}$$

F score (Eq. 5) is a metric calculated as the harmonic mean of precision and recall and, thus, allows to describe the influence of false positives and false negatives at the same time to judge the effectiveness of the classification scheme.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

Two methods for pattern generation were used. The first pattern generation method requires the user to train the system (to provide all calibration data) and resulted in a self-generated repetition pattern. It is used to evaluate how well the algorithm is able to perform with user's own training data, targeting the scenario of users doing their first resistance training session under the supervision of a human personal trainer. Therefore, each user's own calibration data were used as a pattern for detection of repetitions. The second pattern generation method relies on provisioning of the pattern from an external source and results in an *externally generated repetition pattern*. This pattern type is used to test the ability of the algorithm to recognize repetitions using calibration data collected by other users. Thus, the second method reflects the scenario of an autonomous digital trainer with repetition patterns

[1] Although 3,600 exercise repetitions should have been collected, some users made a mistake counting the repetitions, therefore a slightly lower total count of usable repetitions was produced.

Table 3 Repetition detection statistics (average and standard deviation) for externally generated repetition patterns in terms of F score (F), precision, and recall for the constrained and unconstrained environment

Muscle	Constrained			Unconstrained		
	F	Precision	Recall	F	Precision	Recall
Biceps brachii	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Gluteus maximus	0.995 ± 0.016	1 ± 0	0.99 ± 0.031	0.963 ± 0.08	1 ± 0	0.938 ± 0.121
Hamstrings	1 ± 0	1 ± 0	1 ± 0	0.999 ± 0.008	1 ± 0	0.998 ± 0.016
Latissimus dorsi	1 ± 0	1 ± 0	1 ± 0	0.997 ± 0.012	1 ± 0	0.995 ± 0.023
Pectoralis major	0.954 ± 0.067	1 ± 0	0.920 ± 0.114	0.970 ± 0.075	1 ± 0	0.949 ± 0.111
Quadriceps	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Rectus abdominis	1 ± 0	1 ± 0	1 ± 0	0.995 ± 0.016	1 ± 0	0.99 ± 0.03
Soleus	0.999 ± 0.008	1 ± 0	0.998 ± 0.016	0.999 ± 0.008	1 ± 0	0.998 ± 0.016
Triceps brachii	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Average	0.994 ± 0.027	1 ± 0	0.99 ± 0.046	0.991 ± 0.039	1 ± 0	0.985 ± 0.061

prerecorded by, for example, a personal trainer. The results for both pattern generation methods were similar to a great extent; thus, we provide detailed data for the externally generated repetition patterns only.

Repetition detection evaluation yield high precision, recall, and F score values for different exercises and environments. The average F score for all exercises and environments was 0.993 ± 0.034, which is less than 1 % error rate. As Table 3 summarizes, the results show that the number of correctly recognized repetitions was slightly higher for exercises performed in the constrained environment. The slightly worse results in the unconstrained environment are mostly due to lower recall values for the *Gluteus maximus* exercise. One explanation of this observation is that different users performed this exercise in a slightly different but still correct way, which resulted in a higher number of false negatives. However, the difference of the average F scores in the constrained and unconstrained environment is 0.003 % and thus not significant.

5.3 Temporal error of repetition detection

Detecting only the number of correctly recognized repetitions does not offer any means for providing qualitative feedback about the correctness of an exercise. As duration is one of the means to assess whether a training goal has been effectively targeted [14], it can be used to give basic feedback. While the assignment of optimal duration to each exercise is left to training experts, we focus on whether our approach can detect duration accurately enough to give duration-based feedback on exercising performance.

In detail, we analyze how accurately the repetition start time s' and the repetition end time e' can be detected in a correctly classified repetition. To assess s' and e' detection errors, the temporal distances between detected (s', e') tuples and their actual indices in the data stream are observed. Hereby, the actual indices correspond to the timestamps obtained through the process of voice annotation (ground truth used in this study). The average of s' and e' detection errors is calculated for each repetition and

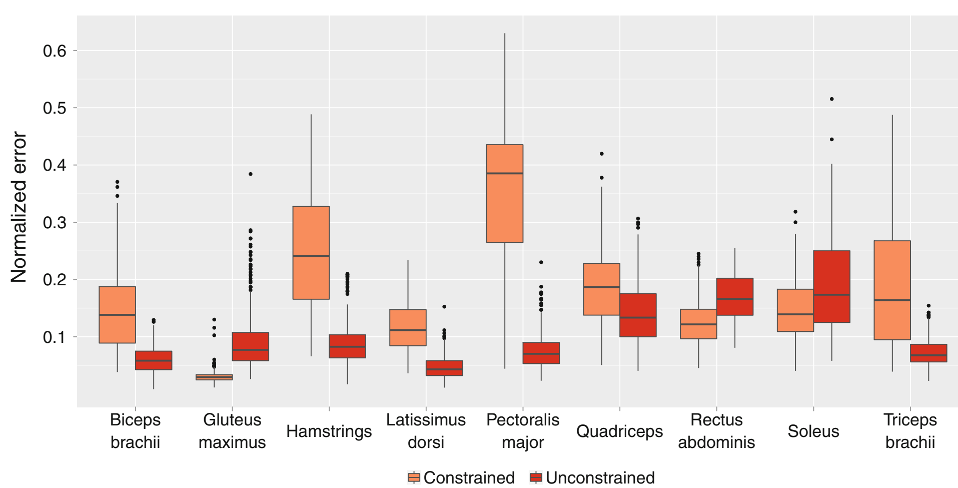


Fig. 5 Temporal error of detecting individual repetition start and end times normalized by the total duration of the repetition for externally generated repetition patterns (median, quartiles, and extreme values)

normalized with respect to the given repetition's duration. Consequently, the evaluation yields a value between 0 and 1 for a given repetition: e.g., value 0 denotes that there was no error detecting the (s', e') tuple and 1 denotes that there was a temporal error of the magnitude of the repetition's duration. Repetitions with a temporal error delay larger than the repetition's duration were counted as incorrectly recognized and, thus, excluded from temporal repetition detection results. The boxplots in Fig. 5 depict normalized repetition start and end time detection errors for different exercises and environments. The overall median temporal error was calculated to be 215 milliseconds, which is around 11 % of the individual repetition's duration. Counterintuitively, the temporal error was higher for most of the exercises in the constrained environment, and almost negligible for some of the exercises in the unconstrained environment. One reason can be seen in the fact that in the unconstrained environment the smartphone is placed on-body, that is, exposed to more direct movement forces and is consequently producing acceleration patterns of higher intensity with more distinguishable repetition start and end times.

6 Conclusion

In this paper, we described how to leverage off-the-shelf smartphones for assisting resistance training exercises in different environments. We introduced an enhanced dynamic time warping (DTW)-based automatic repetition detection algorithm which uses observed acceleration data captured during exercising. The approach has been prototypically implemented on an Android smartphone and tested in an experimental study both indoors with weight machines (constrained environment) and in mostly outdoor scenarios with more freedom to move targeting free weight and resistance band exercises (unconstrained environment).

We provided results in particular for the scenario of training patterns generated by a training expert different to the subject under test. The proposed algorithm achieved a promising overall F score with below 1 % classification error rate while remaining computationally inexpensive. This is achieved by early filtering of sequences which are very unlikely to be repetition sequences. Further, the accuracy of detecting exercise durations resulted in an error with median of about 11 % of the average duration.

Acknowledgments The work of I. Pernek has been supported by Slovenian Research Agency under grant 1000-09-310292 and by Slovene Human Resources Development and Scholarship Fund under grant 11012-34/2010. Parts of the work of K.A. Hummel have been supported by the Commission of the European Union under the FP7 Marie Curie IEF program contract PIEF-GA-2010-276336 MOVE-R.

References

1. ABI (2010) Mobile device user interfaces. Tech rep, ABI Research Report, Q3
2. Ahtinen A, Isomursu M, Huhtala Y, Kaasinen J, Salminen J, Häkkilä J (2008) Tracking outdoor sports—user experience perspective. In: European conference on ambient intelligence, AmI 2008, pp 192–209
3. Alexander JL (2002) The role of resistance exercise in weight loss. Strength Cond J 24(1):65–69
4. Annesi JJ (1998) Effects of computer feedback on adherence to exercise. Percept Mot Skill 87(2):723–730
5. Arzeno N, Deng ZD, Poon CS (2008) Analysis of first-derivative based QRS detection algorithms. IEEE Trans Bio Med Eng 55(2):478–484
6. Asselin R, Ortiz G, Pui J, Smailagic A, Kissling C (2005) Implementation and evaluation of the personal wellness coach. In: 5th international workshop on smart appliances and wearable computing, ICDCSW 2005, pp 529–535
7. Baechle T, Earle R, (U.S.), N.S..C.A. (2008) Essentials of strength training and conditioning. Human Kinetics, Champaign, IL
8. Booth FW, Gordon SE, Carlson CJ, Hamilton MT (2000) Waging war on modern chronic diseases: primary prevention through exercise biology. J Appl Physiol 88(2):774–787
9. Buttussi F, Chittaro L (2008) Mopet: a context-aware and user-adaptive wearable system for fitness training. Artif Intell Med 42(2):153–163
10. Chang KH, Chen MY, Canny J (2007) Tracking free-weight exercises. In: 9th international conference on ubiquitous computing, UbiComp 2007, pp 19–37
11. Delavier F (2010) Strength training anatomy. Sports anatomy. Human Kinetics, Champaign, IL
12. Figo D, Diniz PC, Ferreira DR, Cardoso JaM (2010) Preprocessing techniques for context recognition from accelerometer data. Personal Ubiquitous Comput 14(7):645–662
13. Fitlinxx: http://www.fitlinxx.net/. Last visited: 30 Oct 2012
14. Fleck S, Kraemer W (2004) Designing resistance training programs. Human Kinetics, Champaign, IL
15. Giorgino T (2009) Computing and visualizing dynamic time warping alignments in *R*: the dtw package. J Stat Softw 31(7):1–24
16. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD explor. Newsl. 11(1)
17. Hass CJ, Feigenbaum MS, Franklin BA (2001) Prescription of resistance training for healthy populations. Sports Med 31(14):953–964
18. Jefit: http://www.jefit.com/. Last visited: 30 Oct 2012
19. Liu J, Zhong L, Wickramasuriya J, Vasudevan V (2009) uWave: accelerometer-based personalized gesture recognition and its applications. Pervasive Mob Comput 5(6):657–675
20. Mattmann C, Amft O, Harms H, Troster G, Clemens F (2007) Recognizing upper body postures using textile strain sensors. In: 11th IEEE international symposium on wearable computers, ISWC 2007, pp 1–8
21. Melzi S, Borsani L, Cesana M (2009) The virtual trainer: supervising movements through a wearable wireless sensor network. In: 6th IEEE communications society conference on sensor and AdHoc communications and networks, SECON workshops 2009, pp 1–3
22. Muehlbauer M, Bahle G, Lukowicz P (2011) What can an arm holster worn smart phone do for activity recognition? In: 15th annual international symposium on wearable computers, ISWC 2011, pp 79–82

23. Muscillo R, Conforto S, Schmid M, Caselli P, D'Alessio T (2007) Classification of motor activities through derivative dynamic time warping applied on accelerometer data. In: 29th annual international conference of the IEEE engineering in medicine and biology society, EMBS 2007, pp 4930–4933
24. Page P, Ellenbecker T (2010) Strength band training. Human Kinetics, Champaign, IL
25. Pedersen BK, Saltin B (2006) Evidence for prescribing exercise as therapy in chronic disease. Scand J Med Sci Sports 16(S1):3–63
26. Pollock ML, Franklin BA, Balady GJ, Chaitman BL, Fleg JL, Fletcher B, Limacher M, Pia IL, Stein RA, Williams M, Bazzarre T (2000) Resistance exercise in individuals with and without cardiovascular disease: benefits, rationale, safety, and prescriptionan advisory from the committee on exercise, rehabilitation, and prevention, council on clinical cardiology, american heart association. Circulation 101(7):828–833
27. R Development Core Team (2008) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org. Last visited 30 Oct 2012
28. Roberts CK, Barnard RJ (2005) Effects of exercise and diet on chronic disease. J Appl Physiol 98(1):3–30
29. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech 26(1):43–49
30. Sato K, Smith SL, Sands WA (2009) Validation of an accelerometer for measuring sport performance. J Strength Cond Res 23(1):341–347
31. Savitzky A, Golay MJE (1964) Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 36(8):1627–1639
32. Schlenk EA, Dunbar-Jacob J, Sereika S, Starz T, Okifuji A, Turk D (2000) Comparability of daily diaries and accelerometers in exercise adherence in fibromyalgia syndrome. Meas Phys Educ Exerc Sci 4(2):133–134
33. Seeger C, Buchmann A, Van Laerhoven K (2011) myHealthAssistant: a phone-based body sensor network that captures the wearer's exercises throughout the day. In: 6th international conference on body area networks, BodyNets 2011, pp 1–7
34. Speck BJ, Looney SW (2001) Effects of minimal intervention to increase physical activity in women: daily activity records. Nurs Res 50(6):374–378
35. Tormene P, Giorgino T, Quaglini S, Stefanelli M (2009) Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artif Intell Med 45(1):11–34
36. Warburton DE, Nicol CW, Bredin SS (2006) Health benefits of physical activity: the evidence. Can Med Assoc J 174(6):801–809