

RESEARCH

Open Access



Exploring smartphone sensors for fall detection

Isabel N. Figueiredo^{1*}, Carlos Leal¹, Luís Pinto¹, Jason Bolito² and André Lemos²

*Correspondence:

isabelf@mat.uc.pt;

<http://www.mat.uc.pt/~isabelf>

¹ CMUC, Department of
Mathematics, University of Coimbra,
Coimbra, Portugal

Full list of author information is
available at the end of the article

Abstract

Falling, and the fear of falling, is a serious health problem among the elderly. It often results in physical and mental injuries that have the potential to severely reduce their mobility, independence and overall quality of life. Nevertheless, the consequences of a fall can be largely diminished by providing fast assistance. These facts have led to the development of several automatic fall detection systems. Recently, many researches have focused particularly on smartphone-based applications. In this paper, we study the capacity of smartphone built-in sensors to differentiate fall events from activities of daily living. We explore, in particular, the information provided by the accelerometer, magnetometer and gyroscope sensors. A collection of features is analyzed and the efficiency of different sensor output combinations is tested using experimental data. Based on these results, a new, simple, and reliable algorithm for fall detection is proposed. The proposed method is a threshold-based algorithm and is designed to require a low battery power consumption. The evaluation of the performance of the algorithm in collected data indicates 100 % for sensitivity and 93 % for specificity. Furthermore, evaluation conducted on a public dataset, for comparison with other existing smartphone-based fall detection algorithms, shows the high potential of the proposed method.

Keywords: Fall detection, Smartphone, Tri-axial accelerometer, Threshold-based method

Introduction

Statistics and facts related with falls in elderly people is somewhat worrying. For instance, approximately one in every three people, over the age of sixty five, experience a fall, at least once a year, and these are the leading cause of hospitalization for this age group [16]. Another very concerning aspect of falls, among the elderly, is their reluctance to seeking treatment after suffering an injury. Moreover, the economic impact of falls was estimated in 2000 to be \$US19 billion in the US only [15]. All of this is even more relevant when one considers that the number of old people (above 60 years old) in the world is expected to increase from 841 million in 2013 to more than 2 billion in 2050 [8].

The previous findings, stress the necessity for healthcare providers to focus on measures to reduce the risk and severity of falls-related injuries. Automatic fall detection systems are an important component in this effort and are a current major research topic. The features generally used for fall detection are the magnitude of the acceleration, posture monitoring, change in orientation, vertical velocity, angular velocity, and angular

acceleration [3, 4, 6, 12, 18]. Automated image analysis systems based on video camera images have also been proposed [13]. Other approaches like the *GoSafe* system (<http://www.lifelinesys.com/content/>), known as *PERS* (personal emergency response system), a commercial wearable device from *Philips*, allow users to push a emergency button in the event of a fall. However, these solutions have some drawbacks. The wearable devices are not well tolerated by the elderly while camera-based solutions can not be applied without violating privacy.

In response to some of these issues, many researchers are focusing their efforts on smartphone-based applications. In fact, the increasing popularity of smartphones makes them an attractive platform for the development of new fall detection systems. Moreover, smartphones are well accepted, even among the elderly population, and the already built-in communication facilities, including, e.g., SMS (short message service) and GPS (global position system), makes them a perfect candidate for an automatic fall detection system that covers the detection and communication stages. The increasing number of built-in sensors, such as accelerometer, gyroscope, and magnetometer, is also highly advantageous to researchers. On the other-hand, smartphone systems also involve many challenges. Issues like the huge variety of devices, and inherently the massive amount of software and hardware, make the task of developing new algorithms a formidable challenge.

Smartphone-based fall detectors have already been presented in the literature [1, 7, 19] and some dedicated applications, like e.g. the *iFall* [14], are available in the Android Play Store. A common aspect in all of these studies is the use of threshold-based algorithms and accelerometer data. For instance, in [7] four thresholds are proposed for the difference between the maximum and minimum values of the magnitude of the acceleration vector and vertical acceleration in four defined time windows. In the algorithm described in [1], a fall is simple defined as a peak in the magnitude of the acceleration higher than a threshold followed by a time period where the magnitude of the acceleration is lower than another threshold. The reason for using threshold methods and acceleration-only features aims at reducing battery drain. Threshold methods have low complexity and computational cost, while the accelerometer has lower power consumption when compared e.g. with the gyroscope. Note that accelerometers are also cheaper than gyroscopes and, therefore, more common in smartphones. For further review on fall detection methods and challenges we refer to [9, 11].

The purpose of this paper is to describe a new fall detection algorithm, as well as the evaluation of its performance in collected data and in a public dataset. The proposed algorithm is devised to detect a fall (a sudden incontrollable descent) suffered by the smartphone user. It relies only on the data information provided by the smartphone built-in accelerometer (it does not use other built-in sensors, such as gyroscopes and magnetic sensors, or other sensors such as camera, proximity sensor and microphone) and in addition it is a threshold-based algorithm. These two characteristics are of utmost importance since they help reducing battery power consumption, a crucial issue for smartphone users. In effect, fall detection algorithms are to be incorporated in a more general assistance system. This system has already a considerable battery power consumption, so it is important to develop fall detection algorithms that are less complex and hence require low computational power, in order to increase the battery life time, which is obviously an advantage for the smartphone user. In addition, the tests conducted for evaluating the algorithm (and reported herein), as well as the currently on going tests in real environments (in

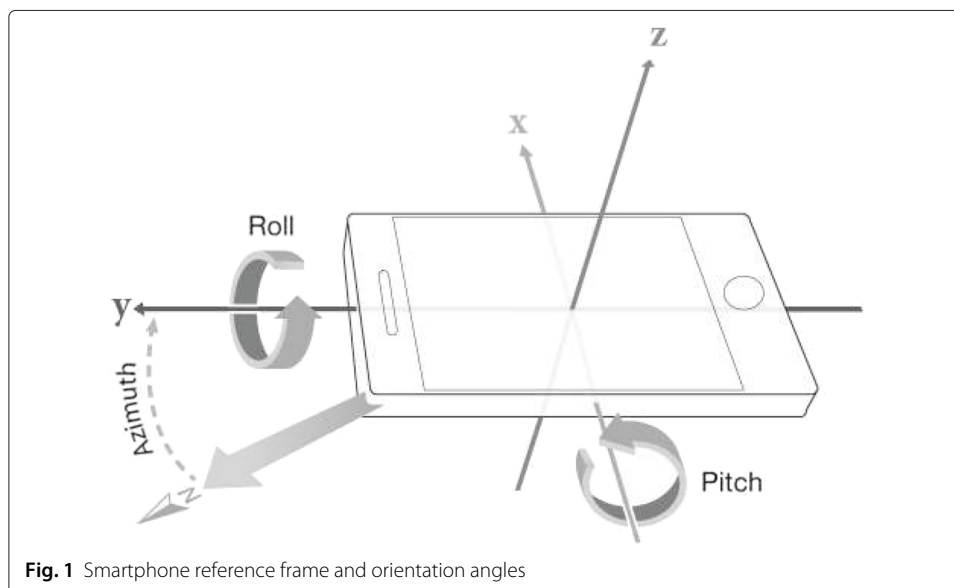
senior care or senior monitoring contexts, both residential or domiciliary) carried out by our collaborative partner (<http://oncaring.com/>) demonstrate its very good performance. Moreover a modified algorithm is also described in order to further increase the battery life-time. Furthermore, experiments conducted on the public dataset MobiFall [17] confirm the good performance of the proposed fall detection algorithm and its superiority over other existing algorithms based on smartphone sensors, for fall detection.

The remainder of the paper is organized in five sections. In Section “Experimental setup”, we detail the material and experimental setup, in Section “Feature extraction” we present the features extracted for analysis, and in Section “Fall detection algorithms and results” algorithms and numerical results are discussed. In Section “Support vector machine classifier” we describe the results obtained with a support vector machine binary classifier. In Section “Comparative study” a comparative evaluation, in a public dataset, with three other fall detection methods is presented. Finally, we draw some conclusions and discuss future research directions.

Experimental setup

The smartphones used were a Samsung Galaxy Nexus and a Samsung Galaxy Nexus S, both equipped with the Android operating system (version 4.1.2). These devices have a wide range of sensors, including triaxial accelerometer, triaxial gyroscope, and magnetometer. In addition to these sensors, Android also provides, e.g., the linear acceleration and the orientation of the device. These quantities are usually obtained through the fusion of sensor data, however, the exact method is not available, and the actual implementation may differ from device to device. For completeness, a very brief description of these sensors and quantities is given herein. We also refer to Fig. 1, where the smartphone reference frame (x -, y - and z - axis) and orientation angles (roll, pitch and azimuth) are depicted.

The accelerometer sensor measures the acceleration force in meters per square second (m/s^2) applied to the device along each axis. We recall that this acceleration signal includes the effect of gravity. The gyroscope returns the angular velocity in radians per



second (*rad/s*), i.e., the rate of rotation around each one of the three axes. The magnetometer measures the magnetic field in microtesla (μT), and it is used to calculate the azimuth. The orientation sensor gives the rotation angles (roll, pitch, and azimuth) of the smartphone with respect to the correspondent axis, and their calculation involves the accelerometer, magnetometer and gyroscope sensors. Finally, the linear acceleration is the acceleration without gravity, also known as dynamic acceleration. This linear acceleration can be obtained by projecting the acceleration into a fixed coordinate system, using the orientation information, and removing the known gravity vector. Alternatively, a high-pass filter can be applied, for deriving the linear acceleration.

The data reported in this subsection were collected by the company *Oncaring* (<http://oncaring.com/>). The data-set consists of simulated falls and ADL (activities of daily living). The simulated falls were performed by two young adults, and for safety reasons a mattress was used. The ADL collection involved six individuals. During these experiments, the smartphone was positioned in the trouser front pocket or in a belt worn around the hip. A total of 74 falls and 136 ADL were recorded overall. Among these, 6 falls and 37 ADL were obtained with the Samsung Galaxy Nexus S. The complete list of ADL activities is given in Table 1. This set of ADL was chosen in such a way that it would be representative of daily activities that can potentially cause false positives in fall detection threshold-based algorithms relying only on the acceleration sensor.

For the movements, the signal of the gyroscope, accelerometer, liner acceleration, and orientation were recorded. The accelerometer and gyroscope amplitude range was set at $\pm 20 m/s^2$ and $\pm 50 rad/s$, respectively. The sampling frequency f was fixed at 100 Hertz (*Hz*) in the Samsung Galaxy Nexus and at 50 *Hz* in the Samsung Galaxy Nexus S. The acquired data is inherently affected by noise measurement. Here, we applied a first-order exponential low-pass filter (cut-off frequency of 5 *Hz*) for smoothing and noise reduction. The choice of this type of filter is motivated by their simplicity and suitability for real-time implementation.

Feature extraction

The chosen features analyzed in this paper for fall detection are described in this section. We remark that these features, which correspond to appropriate quantifications of the sensors' signals, are then used for a binary classification of the different movements into fall or non-fall. This decision is based on a simple thresholding approach, that is afterwards explained in Section "Fall detection algorithms and results", by checking for each movement the different features sequentially.

At time t^n we denote the accelerometer data by $A^n = (A_x^n, A_y^n, A_z^n)$, for $n = 1, \dots, N$, with N the total number of samples, represent by $L^n = (L_x^n, L_y^n, L_z^n)$ the linear acceleration signal, and the output of the orientation sensor, roll, pitch, and azimuth angles, respectively, is represented by $O^n = (O_r^n, O_p^n, O_a^n)$. The measurements of the gyroscope at time t^n are denoted by $G^n = (G_x^n, G_y^n, G_z^n)$. Moreover, we also introduce the vectors

$$A = (A^n)_{n=1}^N, \quad L = (L^n)_{n=1}^N, \quad O = (O^n)_{n=1}^N, \quad \text{and} \quad G = (G^n)_{n=1}^N$$

as well as

$$A_x = (A_x^n)_{n=1}^N, \quad A_y = (A_y^n)_{n=1}^N, \quad A_z = (A_z^n)_{n=1}^N,$$

Table 1 List of ADL movements and simulated falls performed by the volunteers

ADL	Total number
Walk	14
Walk downstairs and upstairs	9
Run	5
Sit down and stand up from a chair	15
Sit down and stand up from the floor	15
Sit down and stand up from a low stool	4
Sit down and stand up from a couch	4
Lie down and stand up from the bed	15
Lie down and stand up from the floor	6
Pick an object off the floor	8
Get in and out of the tub	5
Bump into someone	7
Sit down abruptly on a chair	8
Roll over on the floor	4
Kneel down to the floor	8
Get in and out of the car	5
Vertical jump	4
FALLS	
	Total number
Backward fall ending lying in lateral position	6
Backward fall ending lying	4
Forward fall with arm protection ending lying	7
Forward fall ending lying in lateral position	11
Lateral fall to the right ending lying	4
Lateral fall to the left ending lying	5
Lateral fall against wall ending lying	6
Fall from a chair ending lying	10
Run and fall ending lying	11
Walk and fall ending lying	10

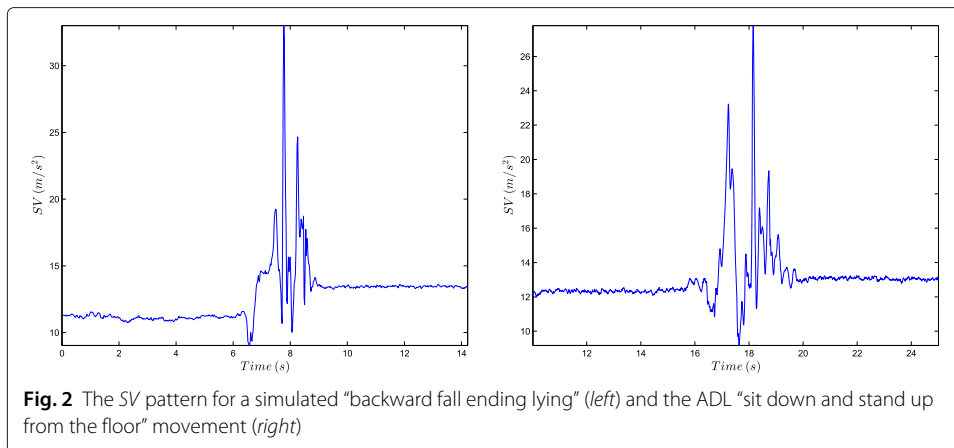
and similar definitions apply to L_x, L_y, L_z , to O_r, O_p, O_a and to G_x, G_y, G_z . In addition we remark that the number of samples can vary with the frequency of the signal.

Feature SV (sum of the components of the acceleration vector) The first feature used to distinguish between fall and ADL is the sum of the absolute value of the components of the acceleration vector (SV), which is defined as follows,

$$SV = (SV^n)_{n=1}^N, \quad SV^n = |A_x^n| + |A_y^n| + |A_z^n|, \tag{1}$$

with $|\cdot|$ representing the absolute value (we remark that the norm in (1) is equivalent to the Euclidean norm $\sqrt{|A_x^n|^2 + |A_y^n|^2 + |A_z^n|^2}$). A typical example of this quantity for a fall event is shown in Fig. 2. The peaks in SV occur as a consequence of the impact of the human body with the ground. In the same figure, we also plot the measured SV for an ADL. Note the small maximum for SV in the latter.

Although SV seems to be an important feature for fall detection it might not be enough for distinguishing correctly a fall from a non-fall. Therefore extra features, as the following ones, are necessary for a more correct classification.

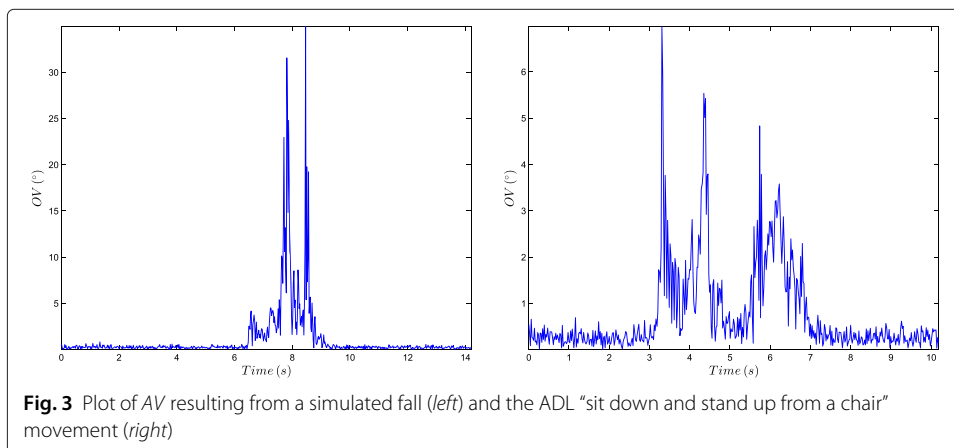


Feature AV (angle variation) It is expected that during a fall event, the acceleration signal will exhibit rapid and significant variations along all the three directions x , y and z . In order to quantify this behavior we define a new fall feature based on the change of the angle between two consecutive acceleration readings. This quantity, herein called orientation variation (OV), is defined by

$$AV = \left[\cos^{-1} \left(\frac{A^n \cdot A^{n+1}}{\|A^n\| \|A^{n+1}\|} \right) \right] \frac{180}{\pi}, \quad \text{for } n = 1, \dots, N - 1, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidian norm, the dot symbol in the numerator stands for the scalar product of two vectors and $\frac{180}{\pi}$ results from radians to degrees conversion. As an example, Fig. 3 shows the feature AV corresponding to the acceleration signal of a simulated fall and a particular ADL, for comparison. The analysis of both figures reveals that during the fall event, AV reaches higher values than during the ADL task. For the fall the maximum value is bigger than 30 degrees (°) and smaller than 7° for the ADL.

Feature CA (change in angle) Another feature that we have studied is the change in orientation of the device before and after a fall. We implement this feature by estimating the angle between two acceleration vectors. First, we define a time window of size 4s



centered at the predicted fall time, that corresponds to the maximum of feature *SV*. Then, we create two vectors, \bar{A}_b and \bar{A}_e , by averaging the acceleration data over the first second and last second of the window of size 4s, respectively. Finally, as in (2), we define the angle

$$CA = \left[\cos^{-1} \left(\frac{\bar{A}_b \cdot \bar{A}_e}{\|\bar{A}_b\| \|\bar{A}_e\|} \right) \right] \frac{180}{\pi}. \tag{3}$$

We expect a significant difference between the initial and final position of the device, and this variation must be reflected in the value of *CA*. Moreover, we take a 4s window because we estimate that a fall lasts approximately 2s.

Feature VA (vertical acceleration) The vertical component of acceleration (*VA*), is a feature usually considered in fall detection algorithms [4, 7]. It is defined by

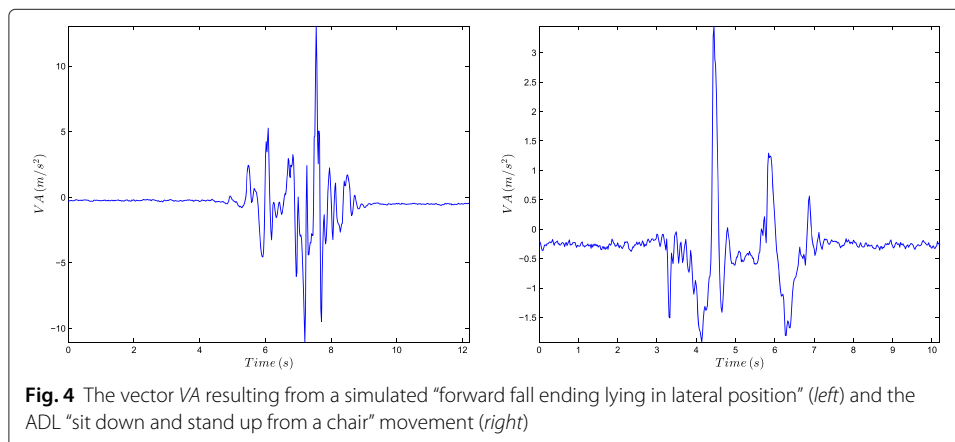
$$VA = \frac{L \cdot g}{\|g\|}, \tag{4}$$

where *g* is the gravity vector, which can be calculated by subtracting the linear acceleration from the acceleration, that is $g = A - L$. *VA* provides a measure for the magnitude of the linear acceleration *L* in the direction of the earth gravity vector. Thus, it is plausible to expect that a fall will produce large values of *VA*, which may be useful to separate fall from non-fall. This prediction is illustrated in Fig. 4 that shows the profile of *VA* during a fall and during an ADL movement.

Feature OD (orientation angles) Now we explore the orientation sensor, by defining a feature that is based on the difference of consecutive values of the sensor output, the roll, pitch and azimuth angles. More precisely, we measure the quantity

$$OD = \frac{1}{\Delta t} \left[|O_r^{n+1} - O_r^n| + |O_p^{n+1} - O_p^n| + |O_a^{n+1} - O_a^n| \right], \quad \text{for } n = 1, \dots, N - 1, \tag{5}$$

where Δt is the inverse of the signal frequency and $|\cdot|$ stands for the absolute value. With this approach we attempt to capture the rapid change in orientation that occurs during a fall. The pitch (O_p) and azimuth (O_a) angles belong to the interval $[-180^\circ, 180^\circ]$ and $[0^\circ, 360^\circ]$, respectively. Note that when a measured value oversteps these limits, a discontinuity or jump of 360° occurs. For instance, the azimuth angle jumps from 0° to 360° (or 360° to 0°) when the measured value is approaching 0° (or 360°). Therefore before



calculating (5), it is necessary to correct the pitch and azimuth angles. Regarding the roll angle it does not show this behavior, and its values change continuously in the interval $[-90^\circ, 90^\circ]$.

Figure 5 displays the OD feature for a fall and an ADL movement, that is similar to an actual fall. In this particular example, the difference in magnitude between the two cases is evident.

Feature SV_G and SV_{GA} (gyroscope) Finally, we consider the information of the gyroscope sensor. What we predict is that a fall leads to significant values in the angular velocity and angular acceleration, when compared to an ADL movement. Moreover, in comparison with ADL tasks, those values should be sufficiently large to provide an accurate distinction between fall and ADL. As mentioned in “Experimental setup” Section, the angular velocity is available as the output of the gyroscope sensor. The components of the angular acceleration, denoted herein by $GA^n = (GA_x^n, GA_y^n, GA_z^n)$ at time n , can be approximated by forward finite differences, that is

$$GA_x^n = \frac{G_x^{n+1} - G_x^n}{\Delta t}, \quad \text{for } n = 1, \dots, N - 1, \tag{6}$$

(similarly definitions apply to GA_y^n and GA_z^n). Based on this reasoning, we study the values of the following two features, the sum of the absolute value of the components of the angular velocity

$$SV_G = (SV_G^n)_{n=1}^N, \quad SV_G^n = |G_x^n| + |G_y^n| + |G_z^n|, \tag{7}$$

and the sum of the absolute value of the components of the angular acceleration

$$SV_{GA} = (SV_{GA}^n)_{n=1}^N, \quad SV_{GA}^n = |GA_x^n| + |GA_y^n| + |GA_z^n|. \tag{8}$$

The SV_G curves displayed in Fig. 6 reveal that, at least in this case, our expectations were confirmed. In the simulated fall, the maximum value of SV_G is bigger than the double of the value of SV_G obtained in the ADL. The analysis of SV_{GA} , not shown here, allows to draw an identical conclusion.

Fall detection algorithms and results

In this section we describe and analyse four fall detection threshold-based algorithms, relying on the features previously defined. We remark that in this work, threshold based

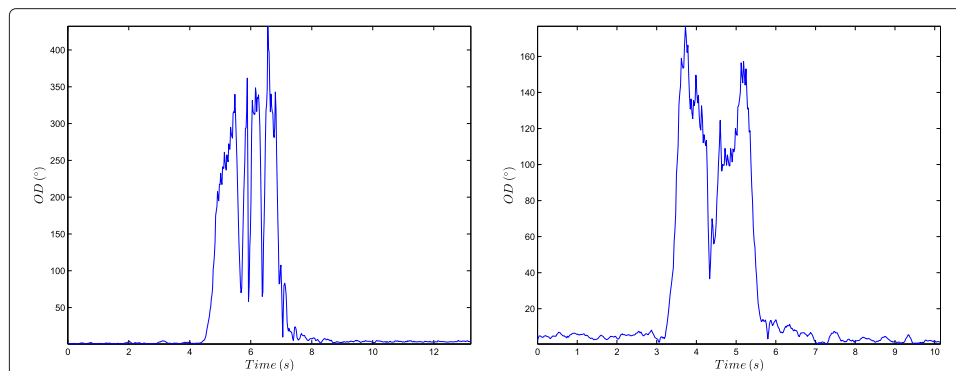
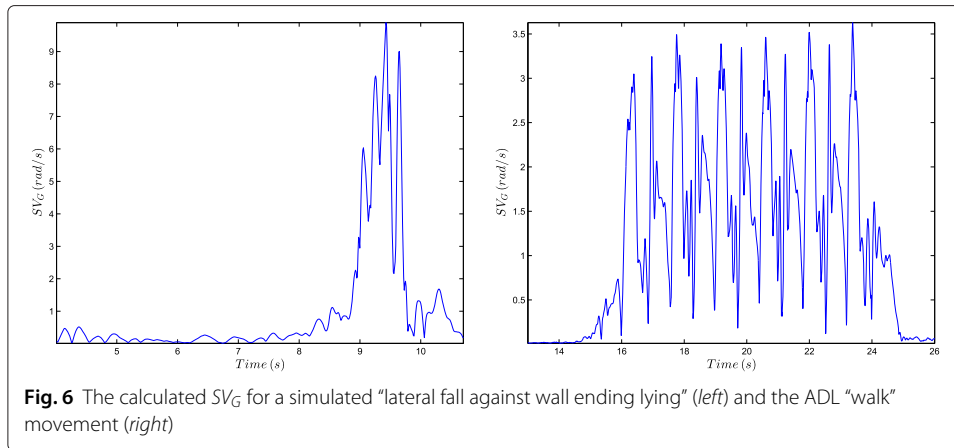


Fig. 5 The vector OD for a simulated “backward fall ending lying in lateral position” (left) and the ADL “pick an object off the floor” movement (right)

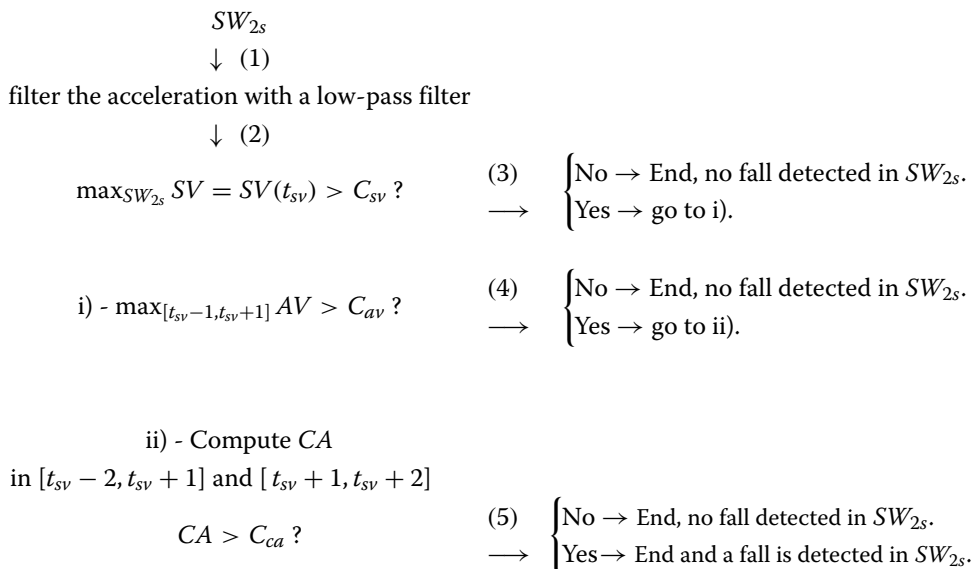


algorithms are chosen, because of their simplicity, low computational cost, which consequently does not increase too much the battery power consumption, a practical and crucial aspect for smartphone users.

Besides the evaluation of the performance of these algorithms, a main goal in this section is also to understand if the use of the linear acceleration, orientation or gyroscope data can significantly improve the performance of the fall detection algorithm that relies only on the acceleration sensor, but somehow incorporates the information about the orientation of the device. Therefore, we consider the following algorithms.

- Alg_1 that only uses acceleration data by combining sequentially the features SV , AV and CA .
- Alg_2 that consists of Alg_1 plus feature VA (the vertical acceleration).
- Alg_3 that consists of Alg_1 plus the orientation feature OD .
- Alg_4 that consists of Alg_1 plus the gyroscope information features SV_G and SG_{GA} .

Flowchart of Alg_1 : Let SW_{2s} be a sliding window of size $2s$ and C_{sv} , C_{av} and C_{ca} be 3 pre-defined and fixed thresholds.



Algorithms Alg_2 , Alg_3 , and Alg_4 correspond to variants of Alg_1 obtained by including extra different features, therefore we omit their descriptions, they are only briefly outlined, because of their similarity with the structure of Alg_1 . In all cases, these extra features are calculated in the time window $[t_{sv} - 1, t_{sv} + 1]$ as in the computation of AV . For Alg_2 , we monitor the maximum value and the l_1 -norm of VA (defined by $\sum_{n=1}^N |VA^n|$) and normalized by the total number of points N , whenever necessary, to account for frequency differences). For Alg_3 we measure the maximum value of OD , and for Alg_4 the maximum values of the features SV_G and SV_{GA} are considered. Moreover, as described above, three thresholds are used in Alg_1 , C_{sv} , C_{av} , and C_{ca} . Besides these three thresholds, we also define the thresholds C_{va} and C_{va}^h for Alg_2 , C_{od} for Alg_3 and C_{svg} and C_{svga} for Alg_4 , with obvious meaning.

All these algorithms were tested on the data-set described in Table 1 of “Experimental setup” Section. The performance of the algorithms was quantified by using sensitivity and specificity measures.

Concerning the choice of the thresholds, they are fixed in order to maximize the sensitivity. Among all the values we choose, for each feature, the maximal value (which is the threshold) such that 100 % of sensitivity is obtained. This high level of sensitivity is essential in fall detection systems. The used threshold values as well as the maximum value for all the falls recorded are shown in Table 2.

The results obtained for each algorithm, in this dataset, are summarized in Table 3 and Table 4 reveal that Alg_1 performs well with 92.65 % for the specificity (which corresponds only to 10 FP). Comparing these results with the performance of algorithms Alg_2 , Alg_3 and Alg_4 it is possible to conclude that apparently the features used in Alg_1 are good enough and no significant advantage is gained by increasing the number of features with extra sensor data.

This conclusion is very important when we take into consideration the computational and power costs. We cannot neglect the limited computational and battery power of smartphone devices. In fact, the power required by the gyroscope, used in Alg_2 , Alg_3 , and Alg_4 , is much higher than the power demand of the accelerometer sensor used in Alg_1 .

To gain more insight about Alg_1 and the influence of the 3 different features SV , AV and CA , we present in Table 5 the sensitivity and specificity results for different features combinations. In particular, we observe that none of these 3 features can be removed without substantially reducing the sensitivity. We also note that feature AV has a strong positive impact in the algorithm performance. To the best of authors knowledge this is the first time that the feature AV is used in a fall detection algorithm.

Table 2 Features and thresholds adopted in the fall detection algorithms

Feature	Fall value	Fall description	Threshold
SV	23.45	Fall from a chair ending lying	$C_{sv} = 23$
AV	18.53	Backward fall ending lying	$C_{av} = 18$
CA	66.33	Forward fall with arm protection ending lying	$C_{ca} = 65.5$
VA	4.93	Fall from a chair ending lying	$C_{va} = 4.72$
VA	1.49	Backward fall ending lying	$C_{va}^h = 1.48$
OD	253.88	Forward fall with arm protection ending lying	$C_{od} = 250$
SV_G	6.02	Backward fall ending lying	$C_{svg} = 5.9$
SV_{GA}	111.21	Backward fall ending lying	$C_{svga} = 110$

Table 3 Sensitivity and specificity results for each algorithm

	TP	FP	TN	FN	Sensitivity	Specificity
Alg_1	74	10	126	0	100 %	92.65 %
Alg_2	74	8	128	0	100 %	94.12 %
Alg_3	74	9	127	0	100 %	93.38 %

With respect to the number of FP of Alg_1 displayed in Table 3, most of them are due to the ADL “sit down abruptly on a chair” movement (4 FP), followed by “get in and out of the car” movement (3 FP). This means, that in more than 50 % of the cases, these two ADL originate FP. These are very similar movements, that can generally be classified as “sitting abruptly”. The pattern for this type of movement resembles a fall, and therefore it poses a real challenge to fall detection systems, especially the ones based solely on acceleration signals. The remaining FP were caused by the ADL “kneel down to the floor” movement (1 FP), and the ADL “lie down and stand up from the floor” movement (2 FP).

We also have tested two algorithms that only use the features SV and VA , and SV and OD , and obtained specificity values of 46.10 and 40.28 %, respectively, which reveals that these features are not reliable. These values can also be compared with those given in Table 5.

The results obtained in this study seem to be in disagreement with some results reported in the literature, but this discrepancy is apparent, and the reason is due to the fact that the other authors use different location for the sensors and also the quality and type of the chosen sensors is different from ours. For instance in [3], 100 % sensitivity and specificity were obtained with an algorithm that is based only on gyroscope data. In our data-set, an algorithm that relies only on SV , SV_G , and SV_{GA} results in 100 % sensitivity and 71.84 % specificity. However, in [3] the authors use an improved gyroscope sensor fixed to the trunk, which is a better place for the sensor in fall detection. We also tried to extract vertical velocity features from the vertical acceleration in a similar way to the one presented in [4]. However, our results were not satisfactory, while in that study the authors also obtained 100 % sensitivity and specificity. But, again, we must stress that in this case better sensors than ours were used, and in addition, the sensors were fixed to the trunk.

Remarks 1. *A requirement demanding at least “24 h” for the minimum duration of the battery life time of the smartphone, was indicated by our industrial partner. Therefore, in order to check whether this requirement was fulfilled, experiments were performed in an affordable device (smartphone) that is also provided by this industrial partner to its senior care assistance services. The battery life time of this smartphone, without the proposed Alg_1 installed, was approximately 28 h (1 day and 4 h). The experiments have shown there was a gain of 50 % in the battery life time if a sampling frequency at 50 Hz is used, with respect to the use of 100 Hz. Due to these results, and the fact that it was also observed that the accuracy of the fall detection with the proposed method did not decrease significantly by*

Table 4 Sensitivity and specificity results for Alg_1 and Alg_4 , in a subset of the dataset that has the gyroscope information (some data do not have the gyroscope information)

	TP	FP	TN	FN	Sensitivity	Specificity
Alg_1	68	8	91	0	100 %	91.92 %
Alg_4	68	6	93	0	100 %	93.94 %

Table 5 Sensitivity and specificity results for a shorter Alg_1 obtained by considering only two different combinations among the 3 features SV, AV and CA

Features	TP	FP	TN	FN	Sensitivity	Specificity
SV + AV	74	38	98	0	100 %	72.06 %
SV + CA	74	56	80	0	100 %	58.82 %
AV + CA	74	19	117	0	100 %	86.03 %

the sampling frequency being fixed at 50 Hz instead of 100 Hz, the decision was to adopt the sampling frequency at 50 Hz in real-life experiments.

Remarks 2. *Finally we remark that we have conducted some experiments using a modified version of the fall detection algorithm Alg_1 , with the goal of decreasing the smartphone battery consumption. The changes are listed below:*

1. *No low-pass filter is applied to the original acceleration signal.*
2. *In step 4 of algorithm Alg_1 the maximum value of AV is computed in the time interval centered in t_{sv} with size 0.7s, instead of 2s as indicated before.*
3. *For the computation of feature CA the time interval $[t_{sv} - 2, t_{sv} + 2]$ of size 4s, is replaced by a shorter time interval also centered in t_{sv} , but with size 3s, and the averages \bar{A}_b and \bar{A}_e are created by averaging the acceleration data over, respectively, the first half second and last half second of this new window of size 3s.*

The modified Alg_1 with the changes 1, 2 and 3 produces a gain in the battery life time of approximately 33 %, with respect to the original Alg_1 (we remark that by removing the low-pass filter the values for feature SV increase, therefore the value C_{sv} was changed to 39). However, the experimental results reveal that the performance is a little bit worse as shown in Table 6 (compare with Table 3). We observe that if we keep the low-pass filter the results are very similar to those obtained with Alg_1 in Table 3, and with the changes 2 and 3, only, there was not a significant increase in the battery life time.

Support vector machine classifier

In this section, we briefly discuss the results obtained with a support vector machine (SVM) binary classifier on our dataset and compare SVM and Alg_1 performances. For this SVM classifier, we use exactly the same features of Alg_1 .

Before proceeding, we observe that in our study we did not consider the possibility of implementing this type of SVM algorithm for fall detection in smartphones, because as it requires more computational power than the threshold-based algorithms, it causes some technical difficulties associated with the lack of energy and computing power in smartphones devices. One possible solution to this problem would be to transmit the data (or at least some of the data), in real time, to a remote computer to perform further analysis. This procedure would allow the use of more sophisticated classification methods like, for example, SVM, whose results are herein discussed.

Table 6 Sensitivity and specificity results for the modified fall detection algorithm Alg_1

Modified Alg_1	Sensitivity	Specificity	TP	FP	TN	FN
Changes 1, 2 and 3	98.65 %	82.35 %	73	24	112	1
Changes 2 and 3	98.65 %	91.91 %	73	11	125	1

For this experiment, the SVM Toolbox LIBSVM developed for Matlab software [5], and the RBF (Radial Basis Function) kernel were adopted. In order to avoid over-fitting, two-fold cross validation was employed. Therefore, our database was partitioned into two sets. Each set contained equal number of falls and ADL, 37 and 68, respectively, and an attempt was made to evenly distribute all the falls and ADLs across both sets.

Two parameters need to be optimized for SVM, a regularizing parameter, and another one that is involved in the definition of the Gaussian (radial basis function) kernel. We have followed the recommended “grid-search” procedure. Thus several pairs of these two parameters were tried, and the one with the best accuracy for the cross-validation of the testing sets was selected.

The classification performance was evaluated by the accuracy measure. The best accuracy results for the set of 3 features $SV + AV + CA$, and also for other combinations, with only 2 out of these 3 features, are given in Table 7.

We point out the performance of the combination $AV + CA$ is very similar to the one obtained with $SV + AV + CA$. In fact, the same sensitivity was obtained and the specificity value is only slightly lower. These results confirm the quality of the three features, especially AV and CA , to accurately classify fall and ADL events.

Finally, in Table 8, we reveal the number of actions that were misclassified by SVM using the three features. It is interesting to note, that among the 5 FP, 4 are “sit down abruptly on a chair” movements, and 1 is the ADL “kneel down to the floor” movement. These 5 actions, with the exception of 1 ADL “sit down abruptly on a chair” movement, were already misclassified by the threshold-based approach Alg_1 . The 3 FN correspond to “a lateral fall to the right ending lying”, “a forward fall with arm protection”, and “a walk and fall ending lying”.

By comparing these results with those obtained for Alg_1 in Section “Fall detection algorithms and results”, we conclude that they are very similar, and consequently the threshold-based algorithm Alg_1 , although simpler has a performance as good as the more elaborated and complex SVM algorithm.

Comparative study

The MobiFall dataset [17] is a publicly available dataset, that was built with the objective of testing new methods and doing comparative evaluation among different fall detection algorithms, based on smartphone sensors. Therefore for further evaluating the performance of the methods described herein and for comparison with the performance of other existing fall detection algorithms (precisely those that are chosen in [17] for a comparative evaluation using the MobiFall dataset, and that are based on mobile phones or smartphone devices only), we carried out experiments in the MobiFall dataset [17]. A short description of this dataset is given in the next subsection.

Table 7 Performance of the SVM classifier

Features	Accuracy	Sensitivity	Specificity
$SV + AV + CA$	96.19 %	95.95 %	96.32 %
$SV + AV$	85.24 %	82.43 %	86.77 %
$SV + CA$	88.57 %	85.14 %	90.44 %
$AV + CA$	95.71 %	95.95 %	95.59 %

Table 8 Classification results for SV + AV + CA using SVM

Features	TP	FP	TN	FN
SV + AV + CA	71	5	131	3

The MobiFall dataset

This dataset was built by the Biomedical Informatics & eHealth Laboratory of the Technological Educational Institute of Crete, Greece. It can be downloaded freely (www.bmi.teicrete.gr) and we refer to [17] for a detailed description.

The MobiFall dataset contains data collected by six male and five female subjects, whose ages ranged from 22 to 36. Nine participants performed falls and ADLs, while two performed only falls. Four different types of falls were simulated, and each fall was repeated three times per subject. The ADL collection consists of nine different activities. According to the authors in [17] the ADLs were chosen based on their similarity to actual falls, which may produce false positives. We remark that these ADLs are similar to the ones presented in our dataset (Table 1). In total the dataset contains 132 Falls and 341 ADLs. A description of the Falls and ADLs is given in Table 9.

For each movement the data from the accelerometer, orientation, and gyroscope were recorded. The data were acquired using a Samsung Galaxy S3 smartphone with accelerometer sampling frequency at 100 Hz and amplitude range at $\pm 20 m/s^2$ (we remark that these values are identical to the ones used in our dataset, however a different smartphone was employed). The device was located in a trouser pocket arbitrarily chosen by the subject in any random orientation, but for the falls the device was in the pocket on the opposite side of the falling direction.

Results and discussion

In this section we do a comparative evaluation of the performance of the fall detection algorithms described in this paper and those presented in [17]. First, we give in Table 10 a new set of thresholds for the features used in our algorithms *Alg1*, *Alg3*, and *Alg4*. As before for the dataset described in this paper, and also in [17], this set of thresholds was

Table 9 Falls and ADLs recorded in the MobiFall dataset

ADL	Total number
Standing with subtle movements	9
Normal walking	9
Jogging	27
Continuous jumping	27
Stairs up (10 stairs)	54
Stairs down (10 stairs)	54
Sitting on a chair	54
Step in a car	54
Step out a car	54
FALLS	Total number
Fall forward from standing, use of hands to dampen fall	33
Fall forward from standing, first impact on knees	33
Fall sideways from standing, bending legs	33
Fall backward while trying to sit on a chair	33

Table 10 Thresholds adopted in the MobiFall dataset

Feature	Fall value	Fall description	Threshold
SV	22.14	Fall backward while trying to sit on a chair	$C_{sv} = 22$
AV	6.94	Fall backward while trying to sit on a chair	$C_{av} = 6.9$
CA	21.17	Fall sideways from standing, bending legs	$C_{ca} = 21$
OD	238.43	Fall backward while trying to sit on a chair	$C_{od} = 236$
SV _G	3.82	Fall backward while trying to sit on a chair	$C_{svg} = 3.8$
SV _{GA}	53.65	Fall backward while trying to sit on a chair	$C_{svga} = 53$

chosen in order to maximize the sensitivity. We remark that Alg_2 was not considered in this section since the linear acceleration data is not provided in the MobiFall dataset.

The results are presented in Table 11 and confirm our previously findings, namely, the good performance of Alg_1 , which shows only a small decrease in specificity from 92.65 % (Table 3) to 85.96 %, and the minor improvement obtained by Alg_3 and Alg_4 . Once again we emphasize the much lower computational and power requirements of Alg_1 , that only relies on the accelerometer data, compared to Alg_3 or Alg_4 . Moreover, a comparison with the algorithms analyzed in [17], and whose results are also displayed in Table 11, shows that the performance of Alg_1 (as well as of Alg_3 or Alg_4) is better. In the analysis of these results we must highlight the higher sensitivity of Alg_1 . A high sensitivity is crucial in fall detection algorithms and in this sense Alg_1 clearly outperforms the three algorithms [7, 10, 14], and yet it still maintains a good specificity. In what concerns the FP registered with Alg_1 in MobiFall dataset, more than 40 % correspond to the “Car-step in” ADL, which is in good agreement with the results obtained in our dataset, where also near 50 % of FP were generated by the movement of the type “sitting abruptly” (described before).

We note that according to our experiments sampling frequencies of 50 or 100 Hz seem to be enough for fall detection. We remark that the use of higher frequencies also brings additional computational cost and battery consumption.

Conclusion

In this paper, we have analysed smartphone sensors to determine their reliability to discriminate between falls and ADL. Based on our results, the accelerometer appears to be the most reliable sensor. Using the information provided by this sensor a novel algorithm was proposed and tested. The algorithm Alg_1 is simple and can easily be implemented in smartphones platforms. In our data-set, its performance leads to 100 % sensitivity and 92.65 % specificity. The SVM analysis confirms the good performance of the proposed methodology. A comparative study with the performance of three existing threshold fall detection algorithms (based on smartphone sensors), using the same public dataset, shows that the proposed algorithm is very competitive.

Table 11 Results for each algorithm in the MobiFall dataset

Method	Sensitivity	Specificity
Alg_1	100 %	85.96 %
Alg_3	100 %	86.26 %
Alg_4	100 %	87.72 %
Sporaso and Tyson [14]	55 %	100 %
Dai et al. [7]	39 %	100 %
He et al. [10]	71 %	84 %

In particular, we mention that in a real and blind test (the user did not know the fall detection algorithm, proposed in this paper, was installed in the smartphone), with the duration of one week, a total of 16 FPs were recorded, thus corresponding to a mean of 2 FPs *per* day. This user came from the target group of older users and it was verified that he was carrying the device.

Moreover, we also observe that it is always possible to create a mechanism in the smartphone that permits the user to disable the alarm signal in case of an FP. However, this is feasible only if the number of FPs is low, otherwise it might stress the user. Additionally, the existence of an alarm signal (for instance a sound alarm), that is activated whenever a fall is detected by the algorithm, can also be understood as a safety issue for the user. In effect in this way the user realizes the system is working.

Furthermore, we emphasize that the particular locations indicated for the smartphone and used in our study (in the trouser front pocket or in a belt worn around the hip) fasten the smartphone to fixed positions of the body's user, that are less prone to random movements. In this way the signals emitted by the sensors will likely lead to better results than those generated by sensors located in regions of the body that are subject to a variety of movements. Hence the detection of falls in other devices, as for instance wristwatches or smartwatches, could be much more difficult, since the arms are parts of the human body that generate many different types of movements. However, the detection of falls using other devices is clearly challenging and may be the subject of future work.

The use of more advanced pattern recognition and machine learning techniques could increase the robustness of the fall detection algorithm. However, such methods are computationally prohibitive for smartphone environments, because of the battery power consumption. In future work we intend to study alternative ways of solving this issue.

The results described herein are promising, but more experiments must be done and other issues should be explored, e.g., the influence of individual physical factors and smartphone location in the body should be further investigated. Furthermore, we also need to address the false positive results generated by the ADL "sitting abruptly" movement. Another, aspect that needs further study is the difference in acceleration signals from simulated and real falls. The study presented in [2] suggests that there are important differences, and the performance of fall detection algorithms with real data is seriously affected. In cooperation with our industry partner we are now testing the algorithm in real world environment. This will allow us to collect valuable data to explore these issues in future research.

Competing interests

Isabel N. Figueiredo, Carlos Leal and Luís Pinto work at the University of Coimbra, Portugal and have no conflict of interest to declare in relation to the presented work. Jason Bolito and André Lemos worked at *OnCaring* (<http://oncaring.com/>), a company for *Senior Care solutions from the Critical Group*, while this work was carried on. *OnCaring* has no commercial relationship with the remaining authors.

Authors' contributions

IF, CL and LP are responsible for the conception and development of new algorithms for detecting falls based on smartphone sensors, as well as, the corresponding implementation of the algorithms in Matlab language. JB and AL contributed to this study and are responsible for the acquisition of data, analysis and interpretation of data, translation of the fall detection algorithms into JAVA language and their installation in the smartphones, and also conducted experimental and real tests with different smartphone users. All authors read and approved the final manuscript.

Acknowledgments

This work was partially supported by the research project PTDC/MATNAN/0593/2012 of FCT (the Portuguese national funding agency for science, research and technology) and by CMUC (the Centre for Mathematics of the University of Coimbra – UID/MAT/00324/2013), funded by the Portuguese Government through FCT/MEC and co-funded by the

European Regional Development Fund through the Partnership Agreement PT2020. The authors would also like to thank Gonçalo Quadros (CEO at *Critical Software*) for having suggested the study of smartphone-based applications, as well as Luís Neves (CEO at *OnCaring*) and João Diogo Ramos (CEO at *Retmarker*) for their important organizational and technical assistance. *OnCaring* collaborated with the academic investigators, provided the smartphones, collected the data, implemented in JAVA and installed in the Android smartphones the proposed algorithms.

Author details

¹CMUC, Department of Mathematics, University of Coimbra, Coimbra, Portugal. ²Oncaring, Coimbra, Portugal.

Received: 19 November 2015 Accepted: 25 April 2016

Published online: 05 May 2016

References

1. Abbate S, Avvenuti M, Bonatesta F, Cola G, Corsini P, Vecchio A (2012) A smartphone-based fall detection system. *Pervasive Mob Comput* 8:883–899
2. Bagala F, Becker C, Cappello A, Chiari L, Aminian K, Hausdorff JM, Zijlstra W, Klenk J (2012) Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS ONE* e37062:7
3. Bourke AK, Lyons GM (2008) A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med Eng Phys* 30:84–90
4. Bourke AK, O'Donovan KJ, ÓLaighin G (2008) The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls. *Med Eng Phys* 30:937–946
5. Chang C-C, Lin C-J (2011) LIBSVM: A library for support vector machines. *ACM Trans Intell Syst Technol* 2:1–27
6. Chen J, Kwong K, Chang D, Luk J, Bajcsy R (2005) Wearable sensors for reliable fall detection. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai. pp 3551–3554. doi:10.1109/IEMBS.2005.1617246
7. Dai J, Bai X, Yang Z, Shen Z, Xuan D (2010) Mobile phone-based pervasive fall detection. *Pers Ubiquit Comput* 14:633–643
8. Department of Economic and Social Affairs Population Division (2013) World Population Ageing 2013. United Nations, New York
9. Habib MA, Mohhtar MS, Kamaruzzaman SB, Lim KS, Pin TM, Ibrahim F (2014) Smartphone-based solutions for fall detection and prevention: challenges and open issues. *Sensors* 14:7187–7208
10. He Y, Li Y, Yin C (2012) Falling-incident detection and alarm by smartphone with multimedia messaging service MMS. *E-Health Telecommunication Syst Networks* 1:1–5. doi:http://dx.doi.org/10.4236/etsn.2012.11001
11. Igual R, Medrano C, Plaza I (2013) Challenges, issues and trends in fall detection systems. *Biomed Eng Online* 12:66. doi:10.1186/1475-925X-12-66
12. Kangas M, Konttila A, Lindgren P, Winblad I, Jämsä T (2008) Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Posture* 28:285–291
13. Leone A, Diraco G, Siciliano P (2011) Detecting falls with 3d range camera in ambient assisted living applications: a preliminary study. *Med Eng Phys* 33:770–781
14. Sposaro F, Tyson G (2009) iFall: An android application for fall monitoring and response. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN. doi:10.1109/IEMBS.2009.5334912
15. Stevens JA, Corso PS, Finkelstein EA, Miller TR (2006) The costs of fatal and non-fatal falls among older adults. *Inj Prev* 12:290–295
16. Tromp AM, Pluijm SMF, Smit JH, Deeg DJH, Bouter LM, Lips P (2001) Fall-risk screening test: A prospective study on predictors for falls in community-dwelling elderly. *J Clin Epidemiol* 54:837–844
17. Vavoulas G, Padiaditis M, Spanakis EG, Tsiknakis M (2013) The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. In: Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on, Chania. pp 1–4. doi:10.1109/BIBE.2013.6701629
18. Wu G (2000) Distinguishing fall activities from normal activities by velocity characteristics. *J Biomech* 33:1497–1500
19. Zhao Z, Chen Y, Wang S, Chen Z (2012) FallAlarm: Smart phone based fall detecting and positioning system. *Procedia Comput Sci* 10:617–624

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com