J Ambient Intell Human Comput
DOI 10.1007/s12652-015-0337-0

ORIGINAL RESEARCH

Falls as anomalies? An experimental evaluation using smartphone accelerometer data

Daniela Micucci[1] · Marco Mobilio[1] · Paolo Napoletano[1] · Francesco Tisato[1]

Received: 3 July 2015 / Accepted: 14 December 2015

Abstract Life expectancy keeps growing and, among elderly people, accidental falls occur frequently. A system able to promptly detect falls would help in reducing the injuries that a fall could cause. Such a system should meet the needs of the people to which is designed, so that it is actually used. In particular, the system should be minimally invasive and inexpensive. Thanks to the fact that most of the smartphones embed accelerometers and powerful processing unit, they are good candidates both as data acquisition devices and as platforms to host fall detection systems. For this reason, in the last years several fall detection methods have been experimented on smartphone accelerometer data. Most of them have been tuned with simulated falls because, to date, datasets of real-world falls are not available. This article evaluates the effectiveness of methods that detect falls as anomalies. To this end, we compared traditional approaches with anomaly detectors. In particular, we experienced the kNN and the SVM methods using both the one-class and two-classes configurations. The comparison involved three different collections of accelerometer data, and four different data representations. Empirical results demonstrated that, in most of the cases, falls are not required to design an effective fall detector.

Keywords Fall detection · Anomaly detection · Novelty detection · Accelerometer data · Smartphone

1 Introduction

Falls are a major health risk that impacts the quality of life of elderly people. When a fall occurs, a prompt notification would help in reducing the injuries that the fall could cause. An effective fall detection system should address the following requirements (Abbate et al. 2012): (1) automatic notification of occurred falls; (2) promptness in order to provide quick help; (3) reliability of the fall detection techniques; (4) communication capabilities in order to alert the caregivers; (5) usability in order to facilitating users' acceptance.

Several solutions have been proposed: some of them addressing the problem as a whole, and others focusing on one specific requirement. The contribution of this article is related to the reliability of the fall detection techniques.

Several factors characterize a fall detection technique: from the sensors used to acquire data, to the features extracted; from the algorithms used to detect falls, to the types of datasets used to train the algorithm. The approaches that have been proposed differ for the choices with respect to those factors.

For what concerns data acquisition, ambient sensors, wearable sensors, or a combination of the two, are the principal data sources used in these techniques (Mubashir et al. 2013; Chen et al. 2012). Many recent approaches investigate the possibility of using the sensors provided by smartphones (Medrano et al. 2014; Sposaro and Tyson 2009; Abbate et al. 2012), which are widespread and require almost no installation or set-up. Moreover, they do not introduce any additional cost, can be used in any place, and are accepted by end users because they are already part of their everyday life.

The techniques also differ in the data type used to detect falls. Most popular fall detection techniques exploit

✉ Daniela Micucci
micucci@disco.unimib.it; daniela.micucci@unimib.it

[1] DISCo, University of Milano - Bicocca, Viale Sarca 336, 20126 Milan, Italy

Published online: 30 December 2015

accelerometer data as the main input to discriminate between falls and activities of daily living (ADL). Figure 1a shows an example of accelerometer data representing a fall that are extracted from the dataset provided by Medrano et al. (2014). In particular, the fall was recorded with a smartphone Galaxy mini. Figure 1b illustrates the accelerometer data recorded by two sensors respectively placed on a Galaxy S II [from the dataset by Anguita et al. (2013)] and a Galaxy Nexus (recorded by ourselves). These data capture the walk performed by two different subjects. It is possible to notice that the captured data share a general trend. This suggests the possibility of defining a method for the detection of falls that can be general and independent from the specific devices.

To verify the effectiveness of the method used by the technique to detect falls, data acquired by the sensors are arranged into labeled datasets containing both ADL and falls, usually simulated by volunteers. Often, datasets are elaborated in order to obtain features: from simple raw data to more complex indicator (e.g., magnitude and Fourier transform) whose processing requires time and computational resources. Methods can be principally divided into two main categories: domain knowledge- and machine learning techniques-based (Mirchevska et al. 2014). The approaches currently proposed, regardless of their classification, have in common the fact that they require a set of falls in their training phase. Unfortunately, human simulations are significantly different from real-world falls (Klenk et al. 2011), and this could make those fall detection techniques not feasible for real-world applications.

For this reason, Medrano et al. (2014) experimented the use of a machine learning technique based on one-class classifier that has only been trained on ADL to detect falls

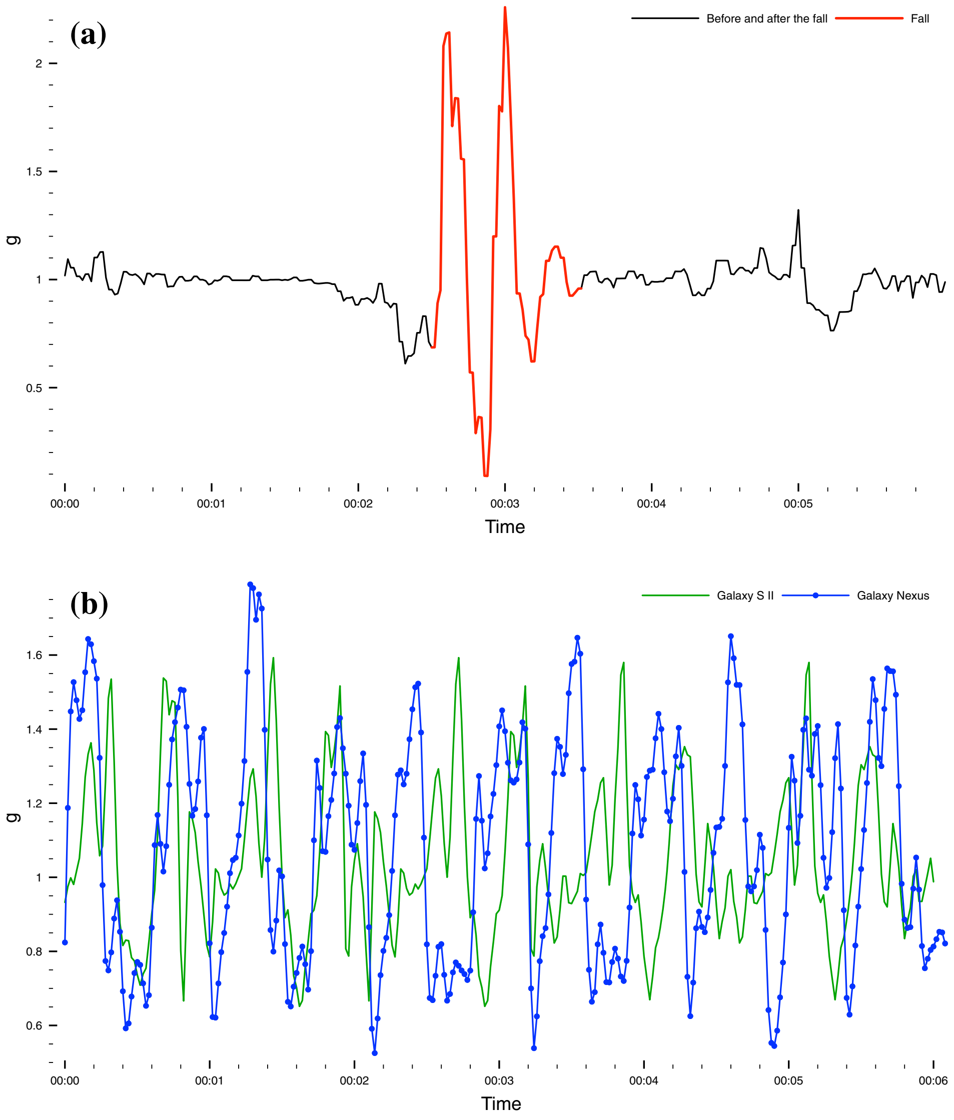

Fig. 1 Examples of accelerometer data: **a** A fall as acquired by a smartphone. **b** A walking activity from two different smartphones performed by two different subjects

as anomalies with respect to ADL. In particular, their experimentation was conducted with a k-Nearest Neighbour (kNN) classifier. As data representation they used the magnitude that does not require an huge amount of resources to be calculated. Medrano et al. (2014) experimented on a publicly available dataset containing both ADL and falls simulated by several human subjects and recorded by the same device. Moreover, Medrano et al. (2014) also experienced a two-classes Support Vector Machine (SVM) on the same dataset. SVM has produced slightly better results with respect to the one-class kNN. Thus, they concluded that anomaly detectors are infeasible in detecting falls. Finally, in the article they explicitly state that data are acquired by accelerometers mounted on smartphones. This suggests that it was taken into consideration the idea of running the analysed methods on smartphones. From our point of view, a smartphone hardly support the execution of a SVM ensuring good performance. Indeed, as Mazhelis (2006) states, SVMs feature very high computational requirements for training. Since the final aim is also to provide a continuous learning system, the high complexity of the training phase is critical when the deployment occurs on a mobile device with limited computational power and, most of all, with limited power resources. Indeed, energy consumption is today one of the main issue in mobile computing, especially when dealing with physical sensors (Pejovic and Musolesi 2015).

The aim of our work is to evaluate the effectiveness of methods that detect falls as anomalies with respect to traditional approaches that use two-classes classifiers to distinguish between falls and ADL. We compared anomaly detectors based on one-class kNN and SVM with traditional detectors based on two-classes kNN and SVM. We considered four different data representations calculated from accelerometer data acquired by smartphones: *raw data*, *magnitude*, *accelerometer features*, and *local temporal patterns*. We evaluated the classifiers with respect to the variations of acquisition conditions: different sensors, different human subjects, different sensor positions. All the experiments have been conducted on two publicly available datasets (Medrano et al. 2014; Anguita et al. 2013) of accelerometer data acquired by smartphones. Evaluation metrics, such as area under the curve (AUC), sensitivity and specificity, confirmed, in most of the cases, that anomaly detection techniques are quite robust against variations of acquisition conditions.

The rest of the paper is organised as follows: Sect. 2 introduces the motivations of our work and discusses related work; Sect. 3 outlines the experiment design; Sect. 4 presents the results of the experimentations; Sect. 5 discusses the achieved results; finally Sect. 6 provides some details about the future directions.

2 Motivation and related work

In the near future the number of elderly people is expected to grow. Indeed, the World Population Ageing Report states that the global share of elderly people (aged 60 years or over) will reach more than 21 % by 2050 (more than 2 billion people) (United Nations 2013). Ageing results from the demographic transition, a process where reductions in mortality are followed by reductions in fertility (United Nations 2013; Carone and Costello 2006). The increasing trend of life expectancy has been directly proportional to the increase in disability (Karmarkar 2009). Thus, oldest people represent the greatest challenge in providing health-related services and identifying ways to assist them in maintaining independence (Mann 2004). Indeed, the 31.2 % of people aged 80–84, and 49.5 % of those over age 85, require assistance with everyday activities (Federal Interagency Forum on Aging-Related Statistics - National Center for Health Statistics 2012). This increment results in a growing need for supports (human or technological) that enable the older population to perform daily activities (US Census Bureau 2013).

Intensive research efforts have been and are still focused on the identification of solutions that from one side automatically assist elderly people in performing daily activities and, on the other side, promptly detect anomalous situations related to diseases or to situations purely related to the old age, such as the worsening of the mild cognitive impairment (Acampora et al. 2013), the prompt identification of conditions favorable to heart failures (Deshmukh and Shilaskar 2015), and the prompt detection of falls (Mubashir et al. 2013).

Falls are a major health risk that impacts the quality of life of elderly people. Among elderly people, accidental falls occur frequently: the 30 % of the over 65 population falls at least once per year; the proportion increases rapidly with age (Tromp et al. 2001). Moreover, fallers who are not able to get up more likely require hospitalization or, even worse, die (Tinetti et al. 1993). Thus, several approaches have been proposed to prompt detect falls. They mainly differ with respect to (1) the sensors used to acquire data, (2) the data representation (features) used by the method, and (3) the method used to detect falls.

Table 1 summarizes the analysis performed on a set of significative approaches. The table has been specifically designed to highlight the characteristic features of each approach in terms of (1) sensors, (2) data representation and (3) methods. In particular, the first two columns show the method and the training set configuration respectively. The third column states whether the approach requires a set of falls to train the algorithm. The fourth column lists the set of features used to infer a fall. Finally, the fifth and

Table 1 Related work

Approach	Method	Falls needed?	Features	Smartphone Ad-hoc	Sensors
Medrano et al. (2014)	K-means + NN	No	Magnitude	Smartphone	Triaxial accelerometer
Tolkiehn et al. (2011)	Threshold based	Yes	Magnitude of standard deviation per axis	Ad-hoc	Triaxial accelerometer
			Std of the magnitude		Barometric pressure
			Ratio of the polar angle		
			Delta of two consecutive polar angles		
			Barometric pressure		
Wang et al. (2014)	Threshold based	Yes	Signal magnitude vector	Ad-hoc	Triaxial accelerometer
			Hearth rate value		Health rate moniter
			Trunk angle		
Bourke et al. (2007)	Threshold based	Yes	Magnitude	Ad-hoc	Dual-axis accelerometers placed orthogonally
Li et al. (2009)	Threshold based	Yes	Magnitude of acceleration	Ad-hoc	Triaxial accelerometer
			Magnitude of angular velocity		Triaxial gyroscope
Zhang et al. (2006)	One-class SVM	Yes	Time of free fall	Ad-hoc	Triaxial accelerometer
			Variance of acceleration during free fall		
			Time of reverse impact		
			Mean and variance of acceleration during reverse impact		
Chen et al. (2006)	Threshold based	Yes	Magnitude	Ad-hoc	Dual-axis accelerometers placed orthogonally
Nyan et al. (2008)	Threshold based	Yes	Correlation coefficient between thigh and waist deviation from vertical axis	Ad-hoc	Triaxial accelerometer
			Correlation coefficient between angular velocity and reference template		Two-axis gyroscope
Abbate et al. (2012)	Threshold based	Yes	Magnitude		Triaxial accelerometer
			Time of inactivity	Smartphone	
			Peak time	Ad-hoc	
			Impact start		
			Impact end		
Ge and Shuwan (2008)	Threshold based	Yes	Inertial frame vertical acceleration	Ad-hoc	Triaxial accelerometer
			Inertial frame vertical velocity		
			Time of free fall		Two-axis gyroscope
Mellone et al. (2012)	Threshold based	Yes	Acceleration sum vector	Smartphone	Triaxial accelerometer
			Vertical axis orientation		
Rabah et al. (2012)	Threshold based	Yes	Magnitude	Ad-hoc	Triaxial accelerometer
			Orientation		

Table 1 continued

Approach	Method	Falls needed?	Features	Smartphone Ad-hoc	Sensors
Shibuya et al. (2015)	SVM	Yes	Range of angular velocity	Ad-hoc	Triaxial accelerometer
			Range of acceleration		Two-axis gyroscope
					Single axe gyroscope
Zhuang et al. (2015)	SVM	Yes	Magnitude	Ad-hoc	Triaxial accelerometer
			Ascending coefficient		
			Descending coefficient		
Sposaro and Tyson (2009)	Threshold based	Yes	Magnitude	Smartphone	Triaxial accelerometer
			Angle of body		
Dai et al. (2010)	Threshold based	Yes	Magnitude		
			Ad-hoc feature	Smartphone	Triaxial accelerometer
			Shape Context	Ad-hoc	Magnetometer
			Hausdorff distance		
Lee and Carlisle (2011)	Threshold based	Yes	Magnitude	Smartphone	Triaxial accelerometer
			Mean of magnitude		
Albert et al. (2012)	SVM	Yes	Moment (mean, abs(mean), std, skew, kurtosis	Smartphone	Triaxial accelerometer
			Moments of the difference between successive samples		
			Smoothed root mean squares		
			Extremes (min, max, abs(min), abs(max))		
			Histogram		
			Fourier components		
			Mean magnitude, mean of cross products (xy, xz, yz), abs(mean of cross products)		
Fang et al. (2012)	Threshold based	Yes	Magnitude	Smartphone	Triaxial accelerometer
			Vertical acceleration value		Gyroscope

sixth columns respectively specify the type of wearable sensor used to sense data (ad hoc solutions or smartphone's sensors) and the involved sensors.

Table 1 aims at providing an idea of how many different approaches are proposed. Most of the approaches rely on data coming from ad-hoc wearable sensing devices, only a few on smartphone's sensors. The mainly used sensors are accelerometers. The approaches use features that are very different each other, some of them very complex in terms of computation. Half of the approaches is based on thresholds-based techniques and the other half on machine learning techniques. Finally, most of the approaches are based on methods that require a set of fall to train the underlying algorithm.

Other approaches not outlined in Table 1 can be found in the many surveys dedicated to the fall detection (e.g., Mubashir et al. 2013; Mohamed et al. 2014; Hijaz et al. 2010). Among the others, Bagal et al. (2012) is particularly interesting because it compares the most popular techniques for the identification of falls based on accelerometer data.

2.1 Sensors and data representation

Fall detection methods rely on data acquired by sensing devices. Images, accelerometer data, audio, angular velocities are only a few examples of data. Data are captured by environmental or wearable sensors or by a combination of both (Mubashir et al. 2013; Chen et al. 2012). Ambient sensors introduce many issues such as privacy, installation costs, and invasiveness. Moreover, a person can fall everywhere. Thus, wearable senors are more indicated for the specific application domain. Under the umbrella of wearable sensors fall ad-hoc solutions and smartphones' sensors. Ad-hoc solutions generally include a microcontroller and a set of attached sensors. Such artifacts are then placed in specific area of the body (e.g., wrist, arm, ankle). Thus, they require an explicit acceptance by the elderly people. On the opposite, smartphones are generally present in everyday life. Therefore, the use of smartphones do not require changes in daily habits and do not involve additional costs.

Despite the type of wearable device, most of the approaches use accelerometers, a few accelerometers with gyroscopes. For this reason, our experimentation has considered accelerometers only.

As regards features, Table 1 shows how the various approaches use features of different nature. Therefore, there not exists a common trend. The unique feature that is found with greater frequency is magnitude.

It is possible to notice that some of the used features are generic such as the magnitude, the energy, and the standard deviation. Others are specifically related to the application domain, such as the time of free fall, the time of reverse impact, and the time of inactivity.

Some of them are performing (such as, magnitude and Fourier transform), but require high processing times and/or considerable computing resources with respect to the application domain and, in case of smartphone, to the device on which the features will be calculated. Indeed, timeliness and lightness in the computation are crucial factors so that those features can be used with effectiveness on smartphones.

2.2 Methods

Methods can be divided into domain knowledge- and machine learning techniques-based (Mirchevska et al. 2014): the former usually apply heuristics, while the latter usually rely on the definition of classifiers able to detect falls. From our perspective, regardless of the type, what differentiates the techniques is the need for data representing falls in the data set used to train the algorithm. Most of the proposed approaches require falls in the training data set in order to properly configure their method. Falls are mostly realized relying on volunteers that are asked to perform daily activities (such as, sitting, walking, and so on) and to simulate falls.

Even if the achieved results by those approaches are very promising, it is quite difficult to generalize the results because almost always the experimentation is limited to one ad-hoc data set only. In addition, as stated by Klenk et al. (2011), simulated falls significantly differ from real-world falls. Thus, having simulated falls in the training dataset could lead to realize classifiers that may show different behaviours with real-world falls.

For the above considerations, a method based on the detection of anomalies with respect to ADL may be a better solution for this kind of application domain. Fall detection is not the only case in which the detection of anomalies is the better choice in designing a classifier. There are many other situations in which real-world data are very difficult to achieve: imagine a system able to infer terrorist attacks. Real world training sets are very rare or even not existing. From the related work analysis, only Medrano et al. (2014) have assessed the robustness of one-class classifiers trained with a set of ADL. Indeed, Medrano et al. (2014) agree with us stating that "traditional approaches to this problem suffer from a high false positive rate, particularly, when the collected sensor data are biased toward normal data while the abnormal events are rare".

Medrano et al. (2014) also experimented a two-classes SVM (Support Vector Machine). They concluded that SVM allows to obtain best results with respect to one-class kNN classifier in detecting anomalies. Although the goodness of the results, the training of a SVM would be more expensive in terms of computation resources than the training process of a kNN-based one-class classifier.

3 Experiment design

In this article we focus on methods that detect falls exploiting smartphone accelerometer data. In particular, we evaluated the robustness of anomaly detectors compared to traditional detectors that, in turn, are tuned with fall instances. To this end, we designed several experimental setups by varying both *materials and methods*:

- *Data* we have created three different collections of smartphone accelerometer data by mixing the data of two publicly available smartphone accelerometer data (Medrano et al. 2014; Anguita et al. 2013) that have been recorded by different devices with different setups. We have created two sets of these collections selecting different sizes of time window of the accelerometer data. Experimenting on these collections permits to assess the robustness with respect to changes in acquisition conditions.
- *Feature vectors* we experimented four different feature vectors ranging from the most simple to the most complex ones: *raw data*, *magnitude*, *accelerometer features*, *local temporal patterns*. Assessing the goodness of feature vectors is very meaningful especially in a mobile and real time environment where the computational capacity may be limited.
- *Classification schema* we compared two different classification schemas based on the k-Nearest Neighbour (kNN) classifier (Fix and Hodges 1951): *one-class* (corresponding to the anomaly detector) versus *two-classes*. Moreover, we also compared the results achieved with the *one-class* and the *two-classes* classification schemas based on the Support Vector Machines (SVM) classifier (Cortes and Vapnik 1995).

3.1 Publicly available datasets

One of the considered datasets contains both Activities of Daily Living (ADL) and falls performed by ten participants, 7 males and 3 females, ranging from 20 to 42 years old (Medrano et al. 2014). The ADL set has been recorded during real-life conditions: participants carried a smartphone in their pocket for at least one week to record everyday behaviour. On average, about 800 ADL instances were collected from each subject during this period. Participants simulated eight different types of falls: *forward falls*, *backward falls*, *left and right-lateral falls*, *syncope*, *sitting on empty chair*, *falls using compensation strategies to prevent the impact*, and *falls with contact to an obstacle before hitting the ground*. Participants wore a smartphone in both their two pockets (left and right) and performed the falls on a soft mattress in a laboratory environment. They repeated each fall three times for a total of 24 fall simulations. The dataset contains 503 falls[1] and 7816 ADL.

Accelerations have been recorded through the built-in triaxial accelerometer of a Samsung Galaxy Mini phone running the Android operating system version 2.2. The sampling rate was not stable, with a value of about 45 ± 12 Hz. During the daily life monitoring, whenever a peak in the acceleration magnitude was detected to be higher than 1.5 g (g = gravity acceleration), a new data instance was recorded. Each data instance included information in a time window of 6 s around the peak. During each fall simulation, a 6 s width time window around the highest peak has been recorded. Afterwards, the offset error of each axis was removed and an interpolation was performed to get a sample every 2 ms (50 Hz). We will refer to this set of data as *dataset1*.

The other dataset considered contains only ADL performed by a group of 30 volunteers with ages ranging from 19 to 48 years (Anguita et al. 2013). Each person was instructed to follow a protocol of 6 activities: *standing*, *sitting*, *laying down*, *walking*, *walking downstairs*, and *upstairs*. Each subject performed the protocol twice while wearing a Samsung Galaxy S II smartphone: on the first trial the smartphone was fixed on the left side of the belt, and on the second it was placed by the user himself as preferred. The tasks were performed in laboratory conditions but volunteers were asked to perform freely the sequence of activities for a more naturalistic dataset. The accelerometer signals were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 s and 50 % overlap, thus obtaining 128 readings/window. The total number of accelerometer instances is 10,299. We will refer to this set of data as *dataset2*.

For the analysis presented in this paper, we considered two different sub-windows of the accelerometer patterns taken around the peak. More in details, we considered two sub-windows of:

- 2.56 s corresponding to a vector 128 samples;
- 1 s corresponding to a vector of 51 values.

3.2 Data collections description

As discussed before, we have created three different collections of smartphone accelerometer data by mixing the data of two publicly available smartphone accelerometer

[1] The authors declared that due to technical issues some falls had to be repeated in a few cases, so the number is higher than 24 × 2 × 10.

data (Medrano et al. 2014; Anguita et al. 2013). For the evaluation we used a tenfold cross-validation approach. The three data collections are then composed of tenfolds, each containing 90 % of training data and 10 % of test data. More in details:

- *Collection 1*. ADL: 7035 training and 781 test data. FALL: 453 training and 50 test data. Both ADL and FALL data have been taken from the *dataset1*;
- *Collection 2*. ADL: 7035 training and 781 test data. Half of ADL data have been randomly taken from the *dataset1* and half from the *dataset2*; FALL: 453 training data and 50 test data. All the FALL data have been extracted from the *dataset1*;
- *Collection 3*: ADL: 9270 training data and 1029 test data. All the ADL data have been taken from the *dataset2*; 453 FALL training data and 50 FALL test data. All the FALL data have been extracted from the *dataset1*.

3.3 Feature vectors

As discussed before, we considered four different feature vectors extracted from two windows with different size: 2.56 s corresponding to a vector 128 samples and 1 s corresponding to a vector of 51 values. More in details we considered:

3.3.1 Raw data

This is the simplest representation. Each data instance is composed of the concatenation of the three accelerometer data (x, y, z), one for each direction. We obtain a final feature vector of size 384 for the case of 128 samples and 153 for the case of 51 samples.

3.3.2 Magnitude

This vector of features has been obtained from the three accelerometer data (x, y, z) as follows:

$$M = \sqrt{x^2 + y^2 + z^2}.$$

We obtained a vector of size 128 and another of size 51.

3.3.3 Accelerometer features

These feature vectors have been obtained by concatenating four different features for each direction: *mean of the acceleration values*, *standard deviation of the acceleration values*, *energy of the acceleration*, and *correlation of the acceleration*. The dimension of the final feature vector is of size 12. The *energy of the acceleration* is calculated as follows:

$$Energy = \sqrt{\frac{\sum_{i=1}^{N} |a_{fft_i}|^2}{N}},$$

where N is the number of samples of the acceleration patterns and a_{fft_i} are the fast Fourier transform components of the input patterns. The *correlation of the acceleration* is calculated between couple of directions: x versus y, x versus z, etc.

3.3.4 Local temporal patterns

This feature is the most complex representation. The feature vector is composed of the concatenation of acceleration patterns achieved by comparing the magnitude of each sample (M_s) with the several boosted magnitude values of N neighbour samples (M_i). The boosted magnitude values corresponding to a given neighbour i are achieved by increasing the original magnitude by an increasing decimal factor:

$$M_i^n = n + M_i,$$

with n ranging from 0 to $M_{\max}$. The value $M_{\max}$ corresponds to the nearest decimal value of maximum magnitude. For each sample we have $M_{\max} + 1$ comparisons as results of the following inequality:

$$M_s > M_i^n.$$

The result of each comparison is represented as a binary vector map of size N with 1 indicating if the above inequality is satisfied and 0 if not. All the comparison maps are then summed in order to obtain a single vector map of size N. The number of neighbours has been set to $N = 6$. The final feature vector is obtained by concatenating all the maps achieved for each sample thus obtaining $6 \times 51 = 306$ and $6 \times 128 = 768$.

3.4 Methods and their evaluation

We have considered the one-class k-Nearest Neighbour (kNN) classifier and the one-class SVM classifier as anomaly detection techniques. These classifiers have been trained only with ADL instances and tested with both ADL and FALL instances. Given a test instance, if the anomaly score is higher than a given threshold, the new instance is classified as an anomaly/fall, otherwise is classified as an ADL. By varying the threshold, the receiver operating characteristic curve (ROC) and the area under the ROC curve (AUC) can be obtained. We calculated also a specific value of sensitivity ($SE = \frac{True\ positives}{Positives}$) and specificity ($SP = \frac{True\ negatives}{Negatives}$). These values have been obtained by selecting the point that maximised their geometric mean $\sqrt{SE \cdot SP}$, in a ROC curve averaged over the cross-validation results.

We compared the anomaly detectors with a *two-classes* kNN and a *two-classes* radial basis SVM. These classifiers have been trained and tested with both instances of ADL and FALL. We converted the distances achieved by the kNN into scores ranging from 0 to 1. By thresholding the scores we drawn the ROC curve.

All the algorithms have been implemented in Matlab and tested with a PC equipped with the Ubuntu 14.10 distribution. Regarding the two variants of kNN for each fold of the 10-cross validation we have performed an inner 10-cross validation for choosing the best number of neighbours k. We experimented 10 values of k ranging from 1 to 10, and we used the Euclidean distance as distance measure. Regarding the SVM classifier, we used the built-in Matlab package. For each fold of the 10-cross validation we have performed an inner 10-cross validation for choosing the best regularisation and kernel parameters. The Matlab implementation of the SVM allows to achieve scores as outputs along with decisions. By thresholding these scores we obtain the corresponding ROC curve.

In order to make the results reproducible, the data collections as well as the Matlab code used for the experiments are available on the authors' website.[2]

4 Results

In the Tables 2, 4, 6, and 8 we report the results achieved on the three collections by all the classification schemas and feature vectors in the case of accelerometer data instances made of 128 samples . It is quite evident that the two-class SVM performs better than the other classification schemas especially over the first two collections. It is also quite clear that *raw data* and *energy* feature vectors perform better than the others with the *raw data* being the best. As can be noticed, the *one-class* and *two-classes* kNN achieve close performance in most cases.

It is clear that the *raw data* (the simplest feature vector) is one of the best feature vectors independently of collection and classification schema (see Tables 4, and 6). This result is a quite new to scientific community since the most used features are usually more complex, such as *magnitude* and *energy features*.

The results achieved on the third collection by all the feature vectors and classification schemas suggest that this collection is not challenging. In fact, this collection has been made by using the ADL from *dataset2* and the FALL from the *dataset1*. The results obtained on this collection depends on the fact that the experimentation protocol of the underlying datasets is quite different. In fact, in the case of the *dataset1* the ADL were obtained by recording at least

Table 2 Results obtained by both kNN and SVM schemas on the three collections

C.	Feat.	Class.	AUC	SE	SP	$\sqrt{SE \cdot SP}$
1	RAW	1-knn	0.955	0.950	0.899	0.924
		2-knn	0.969	0.932	0.927	0.929
		1-svm	0.939	0.936	0.867	0.901
		2-svm	**0.989**	**0.964**	**0.981**	**0.972**
	Magn.	1-knn	0.822	0.900	0.656	0.768
		2-knn	0.876	0.834	0.761	0.794
		1-svm	0.765	0.880	0.615	0.735
		2-svm	**0.977**	**0.924**	**0.954**	**0.939**
	Energ.	1-knn	0.811	0.870	0.695	0.777
		2-knn	0.925	0.910	0.890	0.900
		1-svm	0.843	0.856	0.768	0.811
		2-svm	**0.988**	**0.970**	**0.958**	**0.964**
	LTP	1-knn	0.817	0.850	0.668	0.754
		2-knn	0.853	0.876	0.697	0.780
		1-svm	0.793	0.846	0.637	0.734
		2-svm	**0.974**	**0.910**	**0.940**	**0.925**
2	RAW	1-knn	0.977	0.960	0.933	0.947
		2-knn	0.984	0.958	0.947	0.952
		1-svm	0.971	0.942	0.928	0.935
		2-svm	**0.992**	**0.976**	**0.986**	**0.981**
	Magn.	1-knn	0.885	0.910	0.762	0.833
		2-knn	0.928	0.898	0.814	0.854
		1-svm	0.804	0.908	0.661	0.775
		2-svm	**0.984**	**0.944**	**0.957**	**0.950**
	Energ.	1-knn	0.891	0.870	0.803	0.836
		2-knn	0.916	0.860	0.878	0.868
		1-svm	0.918	0.910	0.851	0.880
		2-svm	**0.990**	**0.968**	**0.983**	**0.975**
	LTP	1-knn	0.885	0.910	0.754	0.828
		2-knn	0.917	**0.930**	0.779	0.850
		1-svm	0.842	0.898	0.684	0.784
		2-svm	**0.984**	0.926	**0.956**	**0.941**
3	RAW	1-knn	0.998	0.990	0.984	0.987
		2-knn	0.997	0.996	**0.993**	**0.994**
		1-svm	0.998	0.986	0.987	0.986
		2-svm	**0.999**	**1.000**	0.974	0.987
	Magn.	1-knn	0.991	0.990	0.972	0.981
		2-knn	0.998	0.996	0.992	0.994
		1-svm	0.637	0.996	0.600	0.773
		2-svm	**1.000**	**1.000**	**0.997**	**0.999**
	Energ.	1-knn	0.898	0.890	0.767	0.826
		2-knn	0.930	0.856	0.907	0.880
		1-svm	0.996	0.998	0.974	0.986
		2-svm	**1.000**	**1.000**	**0.999**	**1.000**
	LTP	1-knn	0.988	0.970	0.964	0.967
		2-knn	0.997	0.990	0.980	0.985
		1-svm	0.980	0.944	0.914	0.929
		2-svm	**1.000**	**0.996**	**0.999**	**0.997**

Here the accelerometer data contain 128 samples. Best result for each collection is reported in bold

[2] http://www.sal.disco.unimib.it/research/ambient-assisted-living/.

Table 3 Results obtained by both kNN and SVM schemas on the three collections

C.	Feat.	Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
1	RAW	1-knn	0.980	**0.980**	0.940	0.960
		2-knn	0.983	0.962	0.952	0.957
		1-svm	0.954	0.946	0.903	0.924
		2-svm	**0.986**	0.954	**0.969**	**0.961**
	Magn.	1-knn	0.958	**0.910**	0.923	0.916
		2-knn	0.961	0.910	0.929	0.919
		1-svm	0.911	0.870	0.844	0.857
		2-svm	**0.967**	0.904	**0.948**	**0.926**
	Energ.	1-knn	0.811	0.870	0.695	0.777
		2-knn	0.925	0.910	0.890	0.900
		1-svm	0.843	0.856	0.768	0.811
		2-svm	**0.988**	**0.970**	**0.958**	**0.964**
	LTP	1-knn	0.936	0.860	0.889	0.875
		2-knn	0.942	0.882	0.891	0.885
		1-svm	0.890	0.826	0.820	0.823
		2-svm	**0.959**	**0.890**	**0.923**	**0.906**
2	RAW	1-knn	0.988	0.960	0.978	0.969
		2-knn	0.990	0.964	0.977	0.970
		1-svm	0.979	0.948	0.954	0.951
		2-svm	**0.990**	**0.960**	**0.981**	**0.970**
	Magn.	1-knn	0.970	0.940	0.926	0.933
		2-knn	0.976	0.942	0.938	0.940
		1-svm	0.942	0.894	0.879	0.887
		2-svm	**0.984**	**0.952**	**0.955**	**0.953**
	Energ.	1-knn	0.891	0.870	0.803	0.836
		2-knn	0.916	0.860	0.878	0.868
		1-svm	0.918	0.910	0.851	0.880
		2-svm	**0.990**	**0.972**	**0.988**	**0.980**
	LTP	1-knn	0.958	0.900	0.902	0.901
		2-knn	0.964	**0.938**	0.894	0.915
		1-svm	0.930	0.900	0.832	0.865
		2-svm	**0.980**	0.934	**0.933**	**0.934**
3	RAW	1-knn	0.998	**1.000**	0.974	0.987
		2-knn	0.997	1.000	0.995	**0.998**
		1-svm	0.999	0.986	**0.997**	0.992
		2-svm	**1.000**	0.996	0.985	0.991
	Magn.	1-knn	0.996	0.990	**0.993**	0.991
		2-knn	0.998	0.998	0.991	0.994
		1-svm	0.763	0.998	0.741	0.860
		2-svm	**0.999**	**0.998**	0.991	**0.995**
	Energ.	1-knn	0.898	0.890	0.767	0.826
		2-knn	0.930	0.856	0.907	0.880
		1-svm	0.996	0.998	0.974	0.986
		2-svm	**1.000**	**1.000**	**0.999**	**1.000**
	LTP	1-knn	0.996	0.960	**0.991**	0.975
		2-knn	0.997	0.986	0.987	0.987
		1-svm	0.997	0.986	0.978	0.982
		2-svm	**1.000**	**0.998**	0.989	**0.994**

Here the accelerometer data contain 51 samples. Best result for each collection is reported in bold

Table 4 Average results obtained by both kNN and SVM schemas with respect to the three collections

Features	Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
RAW	1-knn	0.977	0.967	0.939	0.953
	2-knn	0.984	0.962	0.955	0.959
	1-svm	0.969	0.955	0.927	0.941
	2-svm	0.993	0.980	0.980	0.980
Magn.	1-knn	0.900	0.933	0.797	0.861
	2-knn	0.934	0.909	0.855	0.881
	1-svm	0.736	0.928	0.625	0.761
	2-svm	0.987	0.956	0.969	0.963
Energ.	1-knn	0.867	0.877	0.755	0.813
	2-knn	0.924	0.875	0.892	0.883
	1-svm	0.919	0.921	0.864	0.892
	2-svm	0.993	0.979	0.980	0.980
LTP	1-knn	0.897	0.910	0.795	0.850
	2-knn	0.922	0.932	0.819	0.872
	1-svm	0.872	0.896	0.745	0.815
	2-svm	0.986	0.944	0.965	0.954

Here the accelerometer data contain 128 samples

Table 5 Average results obtained by both kNN and SVM schemas with respect to the three collections

Features	Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
RAW	1-knn	0.989	0.980	0.964	0.972
	2-knn	0.990	0.975	0.975	0.975
	1-svm	0.977	0.960	0.952	0.956
	2-svm	0.992	0.970	0.978	0.974
Magn.	1-knn	0.975	0.947	0.947	0.947
	2-knn	0.978	0.950	0.953	0.951
	1-svm	0.872	0.921	0.822	0.868
	2-svm	0.983	0.951	0.965	0.958
Energ.	1-knn	0.867	0.877	0.755	0.813
	2-knn	0.924	0.875	0.892	0.883
	1-svm	0.919	0.921	0.864	0.892
	2-svm	0.993	0.981	0.982	0.981
LTP	1-knn	0.964	0.907	0.927	0.917
	2-knn	0.968	0.935	0.924	0.929
	1-svm	0.939	0.904	0.877	0.890
	2-svm	0.979	0.941	0.948	0.944

Here the accelerometer data contain 51 samples

one week of daily life, while in the case of the *dataset2*, each person was instructed to perform 6 ADL in a laboratory environment. The difference between the ADL data in the two datasets makes the classification problem too easy.

In the Tables 3, 5, 7, and 9 we report the results achieved on the three collections by all the classification

Table 6 Average results obtained by both kNN and SVM schemas with respect to the four data representation

Features	Collect.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
RAW	1	0.963	0.946	0.918	0.932
	2	0.981	0.959	0.949	0.954
	3	0.998	0.993	0.984	0.989
Magn.	1	0.860	0.884	0.746	0.809
	2	0.900	0.915	0.799	0.853
	3	0.906	0.996	0.890	0.937
Energ.	1	0.892	0.901	0.828	0.863
	2	0.929	0.902	0.879	0.890
	3	0.956	0.936	0.912	0.923
LTP	1	0.859	0.871	0.736	0.798
	2	0.907	0.916	0.793	0.851
	3	0.991	0.975	0.964	0.969

Here the accelerometer data contain 128 samples

Table 7 Average results obtained by both kNN and SVM schemas with respect to the four data representation

Features	Collect.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
RAW	1	0.976	0.960	0.941	0.951
	2	0.987	0.958	0.973	0.965
	3	0.998	0.995	0.988	0.992
Magn.	1	0.949	0.898	0.911	0.905
	2	0.968	0.932	0.924	0.928
	3	0.939	0.996	0.929	0.960
Energ.	1	0.892	0.901	0.828	0.863
	2	0.929	0.903	0.880	0.891
	3	0.956	0.936	0.912	0.923
LTP	1	0.932	0.865	0.881	0.872
	2	0.958	0.918	0.890	0.904
	3	0.998	0.982	0.986	0.984

Here the accelerometer data contain 51 samples

Table 8 Average results obtained by both all classifiers with respect to the four data representation and the three collections

Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
1-knn	0.910	0.922	0.822	0.869
2-knn	0.941	0.920	0.880	0.898
1-svm	0.874	0.925	0.790	0.852
2-svm	0.990	0.965	0.974	0.969

Here the accelerometer data contain 128 samples

schemas and feature vectors in the case of accelerometer data made of 51 samples. Here, the *one-class* and *two-classes* classifiers achieve close performance in every cases. It should be noticed that the performance achieved in the case of the 51 samples by all the kNN based solutions is better than the case of 128 samples. Moreover, even in this case, raw data demonstrated to work better than or comparable with more complex feature vectors.

Table 9 Average results obtained by both all classifiers with respect to the four data representation and the three collections

Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
1-knn	0.948	0.927	0.898	0.912
2-knn	0.965	0.934	0.936	0.934
1-svm	0.927	0.926	0.879	0.901
2-svm	0.987	0.961	0.968	0.964

Here the accelerometer data contain 51 samples

Table 10 Results obtained by the best feature vectors in the case of both the two-classes and one-class classifiers

Coll.	Features	Class.	*AUC*	*SE*	*SP*	$\sqrt{SE \cdot SP}$
1	RAW (128)	2-svm	0.989	0.964	0.981	0.972
	RAW (51)	1-knn	0.980	0.980	0.940	0.960
2	RAW (128)	2-svm	0.992	0.976	0.986	0.981
	RAW (51)	1-knn	0.988	0.960	0.978	0.969
3	Energ. (128)	2-svm	1.000	1.000	0.999	1.000
	RAW (51)	1-svm	0.999	0.986	0.997	0.992

The results achieved by the SVM classifier in the case of 128 samples make clear that the *two-classes* classifier performs better than a novelty detector. This is not true in the case of 51 samples where we demonstrated that using raw data the gap between SVM and the novelty detector is very small. This results overcome the results achieved by Medrano et al. (2014). They demonstrated that a *two-classes* SVM is much better than a novelty detector when the accelerometer data is represented as *magnitude* and is composed of 51 samples. This result is more visible looking at the Table 10. This table includes the best results achieved by a two-classes and a one-class classifier. It is quite evident that the novelty detector, based on the kNN classifier, achieves a performance that is about 10 % less than the the two-classes SVM classifier. The ROC curves that compare the best one-class versus the best two-classes classifier performed on the collection 1 and 2 are showed in Fig. 2. Also from these figures is quite evident that the performance of the novelty detector is very close to the one of the two-classes classifier.

5 Discussion and conclusion

In this work we evaluated the robustness of anomaly detectors (*one-class* classifier) compared to that of traditional *two-classes* detection methods that, in turn, are tuned

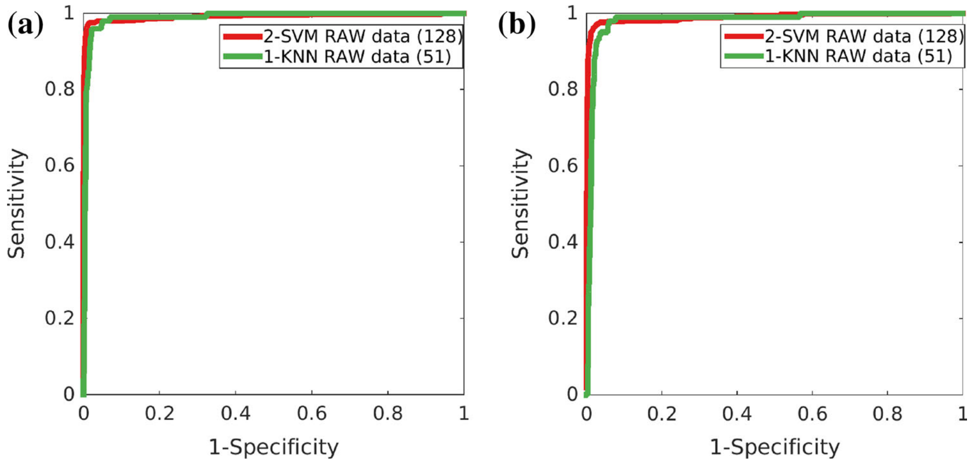

Fig. 2 ROC curves corresponding to the comparison between the best novelty detector and the best two-classes detector. **a** Collection 1. **b** Collection 2.

with fall instances. To this end, we experimented several methods on three different collections of accelerometer data, and four different feature vectors. The experiments have demonstrated that:

- a very simple feature vector based on raw data is very robust to detect falls in both one or two classes schemas;
- a greater number of samples of acceleration instances penalises kNN classification schemas. In contrast, the SVM classifier does not seem to suffer from changes in the number of samples. This makes the one-class kNN classifier more feasible in case of 51 samples;
- in the case of 128 samples a novelty detector is reliable only if it is based on raw data. In the case of 51 samples a novelty detector is reliable if it is based on both raw data and magnitude;

Overall, considering that in the case of raw data, the gap between the SVM and one class kNN is very small, we can conclude that a fall detection system based on a novelty detector is feasible in a real scenario. This is especially true considering the limited computation capacity and power resources of the smartphone. In fact, the raw data does not require further processing and the kNN schema is based on a simple Euclidean distance.

6 Future directions

In order to further validate the robustness of our approach, we should be able to experiment with additional datasets. These datasets should contain ADL performed by different people and recorded by different smartphones.

As the number of data sets freely available is extremely reduced, we decided to develop an application that is able to acquire data from smartphones' sensors and to automatically label them (falls or ADL). This enables us to enrich the datasets of ADL and of simulated falls.

Acknowledgments We would like to thank the Reviewers for their valuable comments and suggestions that allowed us to improve the paper.

References

Abbate S, Avvenuti M, Bonatesta F, Cola G, Corsini P, Vecchio A (2012) A smartphone-based fall detection system. Pervasive Mob Comput 8(6):883–899

Acampora G, Cook DJ, Rashidi P, Vasilakos AV (2013) A survey on ambient intelligence in healthcare. Proc IEEE 101(12):2470–2494

Albert MV, Kording K, Herrmann M, Jayaraman A (2012) Fall classification by machine learning using mobile phones. PLoS One 7(5):e36556. doi:10.1371/journal.pone.0036556

Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: European symposium on artificial neural networks, computational intelligence and machine learning (ESANN)

Bagal 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 7(5):e37062. doi:10.1371/journal.pone.0037062

Bourke A, OBrien J, Lyons G (2007) Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait Posture 26(2):194–199

Carone G, Costello D (2006) Can europe afford to grow old? http://www.imf.org/external/pubs/ft/fandd/2006/09/carone.htm

Chen J, Kwong K, Chang D, Luk J, Bajcsy R (2006) Wearable sensors for reliable fall detection. In: Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS)

Chen L, Hoey J, Nugent CD, Cook DJ, Zhiwen Yu (2012) Sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(6):790–808

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Dai J, Bai X, Yang Z, Shen Z, Xuan D (2010) Mobile phone-based pervasive fall detection. Pers Ubiquitous Comput 14(7):633–643

Deshmukh SD, Shilaskar SN (2015) Wearable sensors and patient monitoring system: a review. In: International conference on pervasive computing (ICPC)

Fang SH, Liang YC, Chiu KM (2012) Developing a mobile phone-based fall detection system on Android platform. In: Computing, Communications and Applications Conference (ComComAp)

Federal Interagency Forum on Aging-Related Statistics - National Center for Health Statistics (2012) Older American 2012: key indicators of well being. http://www.agingstats.gov/agingstatsdotnet/Main_Site/Data/20 12_Documents/docs/EntireChartbook

Fix E, Hodges JL (1951) Discriminatory analysis, nonparametric discrimination: consistency properties. Technical Report, US Air Force School of Aviation Medicine

Ge W, Shuwan X (2008) Portable preimpact fall detector with inertial sensors. IEEE Trans Neural Syst Rehabil Eng 16(2):178–183

Hijaz F, Afzal N, Ahmad T, Hasan O (2010) Survey of fall detection and daily activity monitoring techniques. In: International Conference on Information and Emerging Technologies (ICIET)

Karmarkar AM (2009) Prescription, outcomes, and risk assessment of wheelchairs for aging population. http://d-scholarship.pitt.edu/8349/

Klenk J, Becker C, Lieken F, Nicolai S, Maetzler W, Alt W, Zijlstra W, Hausdorff JM, van Lummel RC, Chiari L, Lindemann U (2011) Comparison of acceleration signals of simulated and real-world backward falls. Med Eng Phys 33(3):368–373

Lee RYW, Carlisle AJ (2011) Detection of falls using accelerometers and mobile phone technology. Age Ageing 40(6):690–696

Li Q, Stankovic JA, Hanson MA, Barth AT, Lach J, Zhou G (2009) Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: International Workshop on Wearable and Implantable Body Sensor Networks (BSN)

Mann W (2004) The aging population and its needs. IEEE Pervasive Comput 3(2):12–14

Mazhelis O (2006) One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection. S Afr Comput J 36:29–48

Medrano C, Igual R, Plaza I, Castro M (2014) Detecting falls as novelties in acceleration patterns acquired with smartphones. PLoS One 9(4):e94811

Mellone S, Tacconi C, Schwickert L, Klenk J, Becker C, Chiari L (2012) Smartphone-based solutions for fall detection and prevention: the FARSEEING approach. Zeitschrift für Gerontologie und Geriatrie 45(8):722–727

Mirchevska V, Lutrek M, Gams M (2014) Combining domain knowledge and machine learning for robust fall detection. Expert Syst 31(2):163–175

Mohamed O, Choi HJ, Iraqi Y (2014) Fall detection systems for elderly care: a survey. In: International Conference on New Technologies, Mobility and Security (NTMS)

Mubashir M, Shao L, Seed L (2013) A survey on fall detection: principles and approaches. Neurocomputing 100:144–152

Nyan M, Tay FE, Murugasu E (2008) A wearable system for pre-impact fall detection. J Biomech 41(16):3475–3481

Pejovic V, Musolesi M (2015) Anticipatory mobile computing: a survey of the state of the art and research challenges. ACM Comput Surv 47(3):47:1–47:29

Rabah H, Amira A, Ahmad A (2012) Design and implementaiton of a fall detection system using compressive sensing and Shimmer technology. In: International Conference on Microelectronics (ICM)

Shibuya N, Nukala B, Rodriguez A, Tsay J, Nguyen T, Zupancic S, Lie D (2015) A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (svm) classifier. In: International Conference on Mobile Computing and Ubiquitous Networking (ICMU)

Sposaro F, Tyson G (2009) iFall: an Android application for fall monitoring and response. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)

Tinetti ME, Liu WL, Claus EB (1993) Predictors and prognosis of inability to get up after falls among elderly persons. JAMA 269(1):65–70

Tolkiehn M, Atallah L, Lo B, Yang GZ (2011) Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)

Tromp AM, Pluijm SM, Smit JH, Deeg DJ, Bouter LM, Lips P (2001) Fall-risk screening test: a prospective study on predictors for falls in community-dwelling elderly. J Clin Epidemiol 54(8):837–844

United Nations (2013) World Populatin Ageing Report 2013. http://goo.gl/z410Ju

US Census Bureau DIS (2013) International programs, international data base. http://www.census.gov/population/international/data

Wang J, Zhang Z, Li B, Lee S, Sherratt RS (2014) An enhanced fall detection system for elderly person monitoring using consumer home networks. IEEE Trans Consum Electron 60(1):23–29

Zhang T, Wang J, Xu L, Liu P (2006) Fall detection by wearable sensor and one-class SVM algorithm. In: Intelligent Computing in Signal Processing and Pattern Recognition, Springer, pp 858–863

Zhuang W, Sun X, Dai D (2015) Fall detection for elder people using single inertial sensor. In: 2015 International Industrial Informatics and Computer Engineering Conference