Triaxial Accelerometer Located on the Wrist for Elderly People's Fall Detection

Armando Collado Villaverde[1], María D. R-Moreno[1], David F. Barrero[1(✉)], and Daniel Rodriguez[2]

[1] Departamento de Automática, Universidad de Alcalá, Crta. Madrid-Barcelona, Alcalá de Henares, Madrid, Spain
david@aut.uah.es

[2] Departamento de Ciencias de la Computación, Universidad de Alcalá, Crta. Madrid-Barcelona, Alcalá de Henares, Spain

Abstract. The loss of motor function in the elderly makes this population group prone to accidental falls. Actually, falls are one of the most notable concerns in elder care. Not surprisingly, there are several technical solutions to detect falls, however, none of them has achieved great acceptance. The popularization of smartwatches provides a promising tool to address this problem. In this work, we present a solution that applies *machine learning* techniques to process the output of a smartwatch accelerometer, being able to detect a fall event with high accuracy. To this end, we simulated the two most common types of falls in elders, gathering acceleration data from the wrist, then applied that data to train two classifiers. The results show high accuracy and robust classifiers able to detect falls.

Keywords: Fall detection · Accelerometer · Machine learning · Classification · Supervised learning · Care for the elderly

1 Introduction

Falls in the elderly are a public health problem [1]. They are not only a significant source of problems associated to elderly for their direct consequences (such as traumas), but also because falls are the symptom of infirmity (such as hearth attack). Therefore, it is not surprising that falls are one of the most relevant concerns for elders care professionals and their families.

The importance of this topic has motivated the rise of a notable number of solution proposals. Most of them have in common the usage of accelerometers [2]. The small size, availability in cell phones and respect to privacy explain why they are becoming so popular in falls detection. Many approaches use dedicated devices, usually placed on the trunk [3], while others exploit the capabilities and popularity of smartphones [4,5]. A similar approach was successfully used to build kinematics models of upper limps [6] and applied to home rehabilitation [7].

Image processing is another popular approach to fall detection [8–10], however, it poses some practical problems which in this context are determinant. People use to dislike having cameras in their private spaces, even if they do

H. Yin et al. (Eds.): IDEAL 2016, LNCS 9937, pp. 523–532, 2016.
DOI: 10.1007/978-3-319-46257-8_56

not record or transmit images. We should also mention the need to install cameras and their limitations to the screened areas. A third group of fall detection systems uses sound or vibrations [11].

Perhaps a notable motivation for elders to reject fall detection devices is their size, which leads to inadequate ergonomics [2]. Fall detection based on cell phones do not present that problem, but raises new ones when they are used by the elderly. Perhaps the most notable one is that they do not keep the cell phone on them when being at home, where most of the falls happen. There are other additional problems, for instance, women use to keep their cell phones in a handbag, where fall detection algorithms will likely fail because they are trained to detect falls through acceleration sensors close to the body trunk.

In order to overcome the usage disadvantages of previous devices, we propose to exploit the popularity of smartwatches. Most of them are programmable devices, and include rich sensing such as accelerometers. Some advanced models even include heart monitors and communication capabilities. All those features together with the price reduction provide a good opportunity of improving elders caring. Some of the problems with cell phones do not happen when using smartwatches: they are located always in the same place, regardless of gender and age, and perhaps more important in elderly people, they are considered every day objects and thus they are not perceived as something invasive. Therefore, we believe these features will help reducing their adoption resistance. In this paper, we present a Machine Learning (ML) -based fall- detection algorithm implemented in a smartwatch. Our results show the accuracy of the classifiers used over the datasets generated under the supervision of professionals in the field.

The rest of the paper is structured as follows. Next, a discussion about the types of falls in elders, then it describes the data acquisition procedure. Section 4 describes the data preprocessing. The main contribution is located in Sect. 5, which describes the training and evaluation of the falls detection algorithm. The robustness of the proposed algorithm is evaluated in Sect. 5.4. The paper finishes with conclusions and future work.

2 Types of Falls in Elders

Our goal is to implement a falls detection algorithm in a smartwatch to monitor the elderly. In order to detect the falls, we pose the problem in terms of classification: given the acceleration values in a time window, classify them as corresponding to a fall or not. Since the final aim is to implement a classifier in a device (a smartwatch) with limited capabilities, the classifier resources consumption is an issue that needs to be taken into account.

As most ML applications, an important issue is how to gather high quality data to train and test the classifiers. In this particular application, data gathering involves people falling, which implies obvious health risks. Other approaches have used volunteers to simulate the fall, who used to be healthy young people, sometimes they were skilled in some martial art or used protections to minimize the risks. This can be a threat the validity as it can suppose a bias to the results, however, data gathering of real falls with elder people is a costly, risky and time consuming task [12], just to mention the most obvious difficulties.

The target of our work is elderly people care, whose falls follow specific dynamics. Elders suffer loose of mobility, which translates to slower motion and increased reaction time. In case of a fall, a young healthy person would react moving his/her arms to cushion the hit; on the contrary, elders do not tend to react in time, resulting in more violent hits. Another relevant issue for our work is the shift of the center of gravity in elders. With age, people tend to separate their legs, and curve the trunk forward, this implies that the center of gravity in elders tend to be lower and shifted forward. As a consequence, falls in elders rarely happen laterally or backwards.

There are usually two types of falls in elders, which we name *syncope* and *forward falls*. We refer to *syncope falls* to those falls consequence of a loss of conscience or hearth condition that prevents using the muscles to control the fall. It results in a vertical motion and a two stages fall: first the knees impact on the ground, and then the trunk moves forward until it hits the ground. The second type of fall usually found in elders is what we name *forward falls*. They are originated by the collision of a foot with an object while the person is walking, loosing the equilibrium and falling over. The trunk in this type of falls moves forward, and given the increased reaction time of elders, the trunk hits the ground without the hands cushioning.

Given the different dynamics of the falls, it seems reasonable to address them as two different, yet related, problems. To this end we will train two different classifiers, each one specialized in detecting one type of fall. In Sect. 5.4 we study the ability of the classifiers to detect falls of a different type they were trained to do so.

3 Data Acquisition

Acquisition of high quality datasets is a key process in ML. We used a smartwatch with a triaxial accelerometer, the hardware imposed a sampling period of 20 ms, yielding three measures of acceleration (X, Y and Z) each 20 ms. The variable Y stands for the longitudinal axis, X for the sideways axis and Z for the axis perpendicular to the watch display.

A key problem to consider is how to build the base class labeled as '*no fall*'. We used data captured along a basket match, removing those samples with accelerations lower than a given threshold. The idea is to keep samples containing high accelerations. This is clearly an unrealistic activity for an elderly person, but basket contains numerous vertical and horizontal motion, making it similar to a fall. In this way we avoid the naïve problem of just classing motion and lack of motion. If a classifier is correctly trained to distinguish between basket and falls, it seems fair to assume that it will be able to distinguish between a fall and normal activities in the life of an elder.

Using real falls was discarded given the risk of injuring the subject, specially when our target is a fragile population group, the elderly. Therefore we tried to capture as much data as realistically possible, taking all necessary measures to avoid risks[1]. We defined two data acquisition procedures, one for syncope and another one for forward falls.

3.1 Syncope Falls Data Capture Procedure

Syncope falls are characterized by the lack of control, with gravity as the only acting force. Other ML approaches to fall detection used volunteers to simulate this type of fall. In our opinion, this scenario is better simulated by using a nursing mannequin, which is a mannequin with the same joints mobility, size and weight than an adult human. Of course, the results will also be biased, since there is only one mannequin available for data gathering, but still the fall is more realistic than a conscious simulating it.

The center of gravity of the mannequin used did not reflect perfectly well the one found in an elderly person. For this reason, just releasing the mannequin does not generate a realistic syncope fall; the mannequin tends to stop once the knees hit the ground. In order to generate realistic falls, it was needed that the mannequin was smoothly pushed forward when it was released. The whole process was supervised by two Geriatrics experts. They helped to optimize the procedure and judged which simulated falls were realistic, and which one should be discarded.

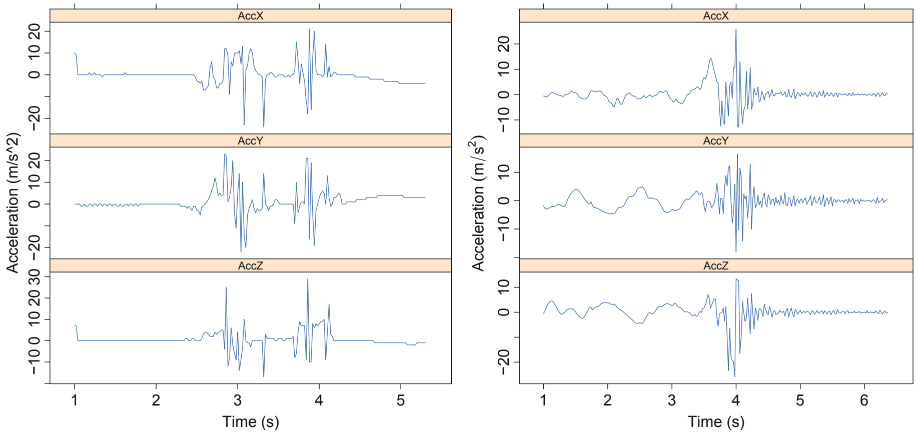
We simulated 42 syncope falls, but the experts only validated 30 falls. There were 12 simulated falls discarded for different reasons, in some cases the mannequin did not fall on its knees, or it hit the ground with the knees, but the trunk did not move forward, or the trunk moved laterally. Falls lasted around 500 ms, but measurements begun shortly before and finished shortly after the fall. Given the fact that each sample contains three acceleration values, and samples are measured in 20 ms intervals, each fall generated between 150 and 300 measures.

3.2 Forward Fall Data Capture Procedure

Forward falls begin with the subject walking, when the subject hits an obstacle overbalancing and falling over. This scenario is poorly approximated with a mannequin. In a forward fall, the subject does have some control, and actually the reflex action is to raise the hands to cushion the hit. In elderly people this reaction can be slow, and they usually do not have enough time to raise their hands, resulting in more dangerous falls. In our opinion, this scenario is better approximated using a healthy young subjects that were trained to move slowly. This is not easy because falling over in that way seems unnatural for the subject, but the results are more accurate according to the geriatric experts consulted.

Therefore, we selected a young volunteer and placed him on a tatami for safety. The obstacle was a thick pad, which also served to safety stop the fall. The domain experts trained the volunteer not to use his hands and move slowly. Once the training was finished, the data capture begun. To simulate the fall,

[1] All datasets are available on http://atc1.aut.uah.es/~david/ideal2016.



(a) Syncope fall acceleration

(b) Forward fall accelerations

Fig. 1. Example of accelerations measured on the wrist.

the volunteer begun to walk and after 4–5 steps hitting the pad with a foot and falling over. Given that data might be affected by which foot hit the pad, we repeated the process the same number of times with each foot.

The volunteer simulated 47 falls, but the experts only validated 40, 20 for each feet. In order to assess the robustness of the classifier, we gathered data from 20 valid falls of two other volunteers. The duration of each fall is around one second.

Figure 1 visualizes the acceleration in a typical (a) syncope and (b) forward fall. In the syncope fall the two hits (knees and trunk) are clearly visible, while the forward fall shows first smooth accelerations due to walking, the fall over and finally the subject laying.

4 Data Preprocessing

Data needs some preprocessing in order to feed the classifier. We grouped data in time windows, which contains samples that serve as input to the classifier. This is also an indirect way to consider history, since the time window contains historical values of acceleration. All windows containing a fall were manually labeled as 'fall', while windows coming from a basket match were labeled as 'not-fall'.

The time window width is a key parameter, we set the width to contain the fall values. We analyzed several syncope falls, observing that the average duration (see Fig. 1) is 500 ms, so we set the time window for syncope fall detection to 500 ms. Similarly, the window length for forward falls was set to 1200 ms.

In addition to raw data coming from the accelerometer, we introduced new attributes summarizing those values. In particular, for each window, we computed the mean and standard deviation of acceleration in each one of the three

Table 1. Attributes under consideration to feed the classifiers: acceleration in X, Y and Z axis along with mean and standard deviation for each axis. N is the number of samples in the window, which depends on the dataset.

Attribute	Label	No. of attributes
Acceleration X	AccelX[X_1, ...,x_N]	N
Acceleration Y	AccelY[y_1, ...,y_N]	N
Acceleration Z	AccelZ[Z_1, ...,z_N]	N
Mean X, Y and Z	MeanX, MeanY, MeanZ	3
Std. deviation X, Y and Z	DevX, DevY, DevZ	3

axis. Those attributes, along with raw data were used to train the classifiers. Table 1 summarizes the attributes considered, however their predictive power greatly varies, as will be analyzed later.

An important issue about data is that it is unbalanced. Since falls are hard to simulate, there were much more data coming from basket than from simulated falls. To face this issue we undersampled the 'not-fall' class, getting the same number of instances for each class. The evaluation of the classifiers was carried out using 10-fold cross-validation.

5 Detection of Fall Events

Fall detection is addressed as a binary classification problem: classify acceleration measures contained in a time window as 'fall' or 'not-fall'. The dataset was build as described in Sect. 3 and then preprocessed to generate time windows, derived attributes and labels as described in Sect. 4.

The goal is to implement the classifier in a smartwatch, which means that the classifier model generated should be as lightweight as possible (memory and computationally). We considered some classical classifiers implemented in Weka such as C4.5 (J48), 1-NN, Logistic regression, Naïve Bayes and PART. Some of these classifiers such as 1-NN are not lightweight, but its good performance motivated us to include them for comparison purposes.

5.1 Determination of Sampling Rate

An important parameter to be set is the sampling rate. Measures were taken with a hardware that imposed a maximum sampling rate of 20 ms, however, it is possible that we could use a lower rate. We should consider that there is a direct relationship between the sampling rate and the number of attributes and therefore trying to reduce the sampling rate might pay off.

We also briefly studied the influence of the sampling rate with the accuracy of the classifiers. To this end we evaluated the accuracy of several classifiers using a range of sample periods. No feature selection was used. The results are shown

Table 2. Evaluation of the influence of the sampling rate on the classifiers accuracy. The table shows the sampling rate, type of fall -syncope (S) or forward (F)-, number of attributes (as indicated in Table 1) and accuracy of the classifiers.

Sample period	Fall type	# attrib	C4.5	1-NN	Reg. Log.	Naïve Bayes	PART
20 ms	S	97	**95.38%**	**97.80%**	**90.26%**	**85.71%**	**95.82%**
40 ms	S	52	92.10 %	97.32 %	87.48 %	84.35 %	91.65 %
60 ms	S	37	88.38 %	92.03 %	87.24 %	83.60 %	87.70 %
20 ms	F	82	**95.66%**	**98.43%**	88.76 %	**86.74%**	**94.09%**
40 ms	F	46	91.97 %	97.21 %	**89.34%**	84.43 %	92.62 %
60 ms	F	34	88.45 %	89.50 %	84.25 %	83.46 %	85.04 %

in Table 2. We can observe a pattern regardless of the classification algorithm, high sampling rates boost performance, at least in the range of values that we have under consideration. The Nyquist Theorem states that there must be a lower bound to the sampling period from which we should not expect further performance improvements. Clearly this lower bound was not achieved as the best performance value was with 20 ms samples. Therefore, this was the chosen value.

5.2 Overview of Attributes Classification Power

The number of attributes used so far is relatively high. To illustrate this point, let us consider a syncope fall window that lasts 500 ms, 25 samples and each sample with three values of acceleration (X, Y, Z), which yields to 75 attributes. In addition, there are six derived attributes (mean and standard deviations), resulting in 81 attributes. This high number of attributes may result in higher computational costs and eventually may also degenerate the classifier performance.

To reduce the number of attributes we estimated the predictive power of the attributes. To this end we applied Correlation Feature Selection Subset Evaluation, as implemented in Weka, This method evaluates the correlation of each attribute with the class, as well as the correlation among the attributes, giving better ranks to those attributes highly correlated with the class while having low correlation among other attributes.

Table 3 ranks attributes by its predictive power suggesting that derived attributes have higher predictive power in comparison to raw acceleration values. In particular, the standard deviation on Y has the best score for syncope falls, and gets the second position for forward falls. Mean acceleration on Z has the second highest score in syncope falls, while appears as a good attribute for forward falls. Raw data seems to have less predictive power, in particular on Z. Interestingly, the mean Z acceleration get a pretty high score for syncope falls, while raw acceleration values on Z does not appear in the table. Syncope fall includes many raw acceleration values on X, with similar scores, this suggests that those attributes may be highly correlated.

Table 3. Twelve best predictive attributes ranked by its correlation to the class for syncope and forward falls.

Forward fall				Syncope fall			
% Inf.	Attrib.	% Inf.	Attrib.	% Inf.	Attrib.	% Inf.	Attrib.
0.515	DevZ	0.173	AccelX1	0.541	DevY	0.233	AccelX11
0.429	DevY	0.163	MeanZ	0.523	MeanZ	0.232	AccelX8
0.399	MeanX	0.159	AccelZ24	0.238	AccelY12	0.229	AccelY11
0.316	DevX	0.154	AccelX2	0.235	AccelX7	0.229	AccelX9
0.209	MeanY	0.144	AccelZ23	0.235	AccelX12	0.227	AccelX10
0.195	AccelX0	0.135	AccelX3	0.233	AccelX13	0.226	AccelX14

Table 4. Performance of syncope (S) and forward falls (F) detection. Attribute selection was performed using a wrapper.

		C4.5	1-NN	Log. Reg.	Naïve Bayes	PART
Accuracy	S	0.98	1	0.92	0.93	0.96
	F	0.98	1	0.89	0.91	0.95
Recall	S	0.98	1	0.94	0.90	0.97
	F	0.98	1	0.91	0.92	0.97
Attributes	S	12	7	16	9	7
	F	9	13	11	12	7

The high score that derived attributes achieve and the hint of highly correlated raw acceleration values suggest that the number of attributes may be reduced significantly. For this reason in the following section we will evaluate the classifiers integrating the feature selection.

5.3 Evaluation of Classifiers

Given the importance of feature selection, we have performed the classifier evaluation along with it using a wrapper approach. This method exploits the interaction between the classifier and the attributes, yielding, in theory, better results, specially where there are a high number of redundant attributes. We used the Weka `WrapperSubsetEval` implementation of the method with Hill Climbing for attribute search.

Table 4 summarizes the performance of the classifiers. For instance, C4.5 (J48 in Weka) with 97 attributes (Table 2) scores 95.38 % accuracy, while using wrapper feature selection it increases to 98 % with only 12 or 9 attributes, depending on the type of fall respectively. The other classifiers behave in a similar way. The 1-NN classifier has an outstanding performance, with a perfect accuracy and recall. The similarity of the instances in the training set may explain this; the

Table 5. Performance of forward falls detection classifiers shown in Table 4 evaluated with a testing set composed by unseen people simulated falls.

		C4.5	1-NN	Log. Reg.	Naïve Bayes	PART
Accuracy	F	0.91	0.66	0.97	0.98	0.98
Recall	F	0.90	0.98	0.95	0.96	0.98
Attributes	F	9	13	11	12	7

robustness analysis done in the next section supports this hypothesis. Despite the magnificent performance of 1-NN, the need to store all the training set dissuades us to implement it in a smartwatch. However, it suggests that perhaps using a Nearest Centroid Classifier may conduct to a good classifier while keeping low computational needs.

5.4 Robustness Analysis of Forward Falls

A clear weakness of the previous approach is the lack of diversity in the training set. Syncope falls used just a single mannequin, while forward falls included falls from one person. This obviously reduces the complexity of the problem, and a natural question that rises is how much the classifier degrades its performance when exposed to different people. In order to provide an insight to this question, we performed a robustness experiment.

Given the lack of alternative mannequin, we focused on forward falls. As described in Sect. 3, we captured data from three people, one of them repeated 40 times the fall simulation, while the others repeated the simulation 20. We exposed the classifiers trained with data coming from the first volunteer (whose performance is shown in Table 4), to the simulated falls of the other two volunteers. The resulting performance is shown in Table 5. As we expected, the performance drops, but in most cases remains above 0.9. The most dramatic case is with 1-NN, whose accuracy falls to 0.66. Logistic regression, Bayes and PART seems quite robust and actually they increase the performance.

6 Conclusions and Future Work

In this paper we have described an ML application to detect falls sensing the acceleration on the wrist. The aim is to implement a fall detection system in a smartwatch oriented to the elderly care. This population is prone to suffer two types of falls, *syncope* and *forward falls*. We simulated those type of falls and measured acceleration on the wrist. These data, along with measures coming from a basket match were used to train and evaluate a classifier with a wrapper-based feature selection.

We selected PART for its high accuracy (above 0.9) and the relatively low number of rules (7) it generated, which made unnecessary the use of external libraries, an interesting feature when looking for a lightweight application. We implemented the algorithm in Android Wear and tested on a Samsung Gear S, with satisfactory results. In a near future we expect to expand the detection with new sensors and an ensemble of classifiers.

Acknowledgements. The authors thank the contribution of Isabel Pascual Benito, Francisco López Martínez and Helena Hernández Martínez, from Department of Nursing and Physiotherapy of the University of Alcalá, for their help designing and supervising the simulated falls procedure. This work is supported by UAH (2015/00297/001), JCLM (PEII-2014-015-A) and MINCECO (TIN2014-56494-C4-4-P).

References

1. Sadigh, S., Reimers, A., Andersson, R., Laflamme, L.: Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a swedish municipality. J. Commun. Health **29**, 129–140 (2004)
2. Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laighin, G.O., Rialle, V., Lundy, J.E.: Fall detection - principles and Methods. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1663–1666 (2007)
3. Gibson, R.M., Amira, A., Ramzan, N., Casaseca-de-la higuera, P., Pervez, Z.: Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. Appl. Soft Comput. J. **39**, 94–103 (2016)
4. Luštrek, M., Kaluža, B.: Fall detection and activity recognition with machine learning. Informatica **33**, 205–212 (2008)
5. Albert, M.V., Kording, K., Herrmann, M., Jayaraman, A.: Fall classification by machine learning using mobile phones. PLoS ONE **7**, 3–8 (2012)
6. Zhou, H., Hu, H.: Reducing drifts in the inertial measurements of wrist and elbow positions. IEEE Trans. Instrum. Measur. **59**, 575–585 (2010)
7. Tao, Y., Hu, H., Zhou, H.: Integration of vision and inertial sensors for 3d arm motion tracking in home-based rehabilitation. Int. J. Robot. Res. **26**, 607–624 (2007)
8. Miaou, S.G., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. In: Conference Proceedings - 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, D2H2 2006, pp. 39–42 (2006)
9. Cucchiara, R., Prati, A., Vezzani, R., Emilia, R.: A multi-camera vision system for fall detection and alarm generation. Expert Syst. **24**, 334–345 (2007)
10. Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion-resistant method based on 3-D silhouette vertical distribution. IEEE Trans. Inf. Technol. Biomed. **15**, 290–300 (2011)
11. Zigel, Y., Litvak, D., Gannot, I.: A method for automatic fall detection of elderly people using floor vibrations and sound-Proof of concept on human mimicking doll falls. IEEE Trans. Biomed. Eng. **56**, 2858–2867 (2009)
12. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J.M., Zijlstra, W., Klenk, J.: Evaluation of accelerometer-based algorithms on real-world falls. PloS one **7**, e37062 (2012)