Evaluation of a Wrist-based Wearable Fall Detection Method

Samad Barri Khojasteh[1,2], José R. Villar[2], Enrique de la Cal[2], Víctor M. González[3], Javier Sedano[4], and Harun Reşit YAZĞAN[1]

[1] Sakarya University, Department of Industrial Engineering,
Sakarya, Turkey
samad.khojasteh@ogr.sakarya.edu.tr,yazgan@sakarya.edu.tr

[2] University of Oviedo, Computer Science Department,
EIMEM, Oviedo, Spain
{villarjose,delacal}@uniovi.es

[3] University of Oviedo, Control and Automatica Department,
EPI, Gijón, Spain
vmsuarez@uniovi.es

[4] Instituto Tecnológico de Castilla y León, Pol. Ind. Villalonquejar,
09001, Burgos, Spain
javier.sedano@itcl.es

Abstract. Fall detection represents an important issue when dealing with Ambient Assisted Living for the elder. The vast majority of fall detection approaches have been developed for healthy and relatively young people. Moreover, plenty of these approaches make use of sensors placed on the hip. Considering the focused population of elderly people, there are clear differences and constraints. On the one hand, the patterns and times in the normal activities -and also the falls- are different from younger people: elders move slowly. On the second hand, solutions using uncomfortable sensory systems would be rejected by many candidates. In this research, one of the proposed solutions in the literature has been adapted to use a smartwatch on a wrist, solving some problems and modifying part of the algorithm. The experimentation includes a publicly available dataset. Results point to several enhancements in order to be adapted to the focused population.

Keywords: Fall detection, Wearable devices, Ambient Assisted Living

1 Introduction

Fall Detection (FD) is a very active research area, with many applications to healthcare, work safety, etc. Even though there are plenty of commercial products, the best rated products only reach a 80% of success. There are basically two types of FD systems: context-aware systems and wearable devices[?]. FD has been widely studied using context-aware systems, i.e. video systems [?]; nevertheless, the use of wearable devices is crucial because the high percentage of elderly people and their desire to live autonomously in their own house [?].

Wearables-based solutions include, mainly, tri-axial accelerometers (3DACC) either alone or combined with other sensors. Several solutions incorporate more than one sensory element; for instance, Sorvala et al [?] proposed two sets of a 3DACC and a gyroscope, one on the wrist and another on the ankle, detecting the fall events with two defined thresholds. The use of 3DACC and a barometer in a necklace was also reported in [?]; similar approaches have been developed in several commercial products.

Several solutions using wearable devices combining 3DACC have been reported, i.e., identifying the fall events using Support Vector Machines [?]. In [?] several classifiers are compared using the 3DACC and the inertial sensor within a smartphone to sample the data. A similar solution is proposed in [?], using some different transformations of the 3DACC signal. The main characteristic in all these solutions is that the wearable devices are placed on the wrist. The reason for this location is that it is much easier to detect a fall using the sensory system in this placement. Nevertheless, this type of devices lacks in usability and the people tend to dismiss them in the bedside table. Thus, this research limits itself to use a single sensor -a marketed smartwatch- placed on the wrist in order to promote its usability.

Interestingly, the previous studies do not focus on the specific dynamics of a falling event: although some of the proposals report good performances, they are just machine learning applied to the focused problem. There are studies concerned with the dynamics in a fall event [?,?], establishing the taxonomy and the time periods for each sequence. Additionally, Abbate et al proposed the use of these dynamics as the basis of the FD algorithm [?]. A very interesting point of this approach is that the computational constraints are kept moderate, although this solution includes a high number of thresholds to tune. Nevertheless, we consider this solution as valid, representing the starting point of this research.

2 Adapting Fall detection to a wrist-based solution

Abate et al [?] proposed the following scheme to detect a candidate event as a fall event (refer to Fig. ??). A time t corresponds to a **peak time** (point 1) if the magnitude of the acceleration a is higher than $th_1 = 3 \times g, g = 9.8m/s$. After a peak time there must be a period of 2500 ms with relatively calm (no other a value higher than th_1). The **impact end** (point 2) denotes the end of the fall event; it is the last time for which the a value is higher than $th_2 = 1.5 \times g$. Finally, the **impact start** (point 3) denotes the starting time of the fall event, computed as the time of the first sequence of an $a <= th_3$ ($th_3 = 0.8 \times g$) followed by a value of $a >= th_2$. The impact start must belong to the interval $[impact\ end - 1200\ ms, peak\ time]$. If no impact end is found, then it is fixed to peak time plus 1000 ms. If no impact start is found, it is fixed to peak time.

Whenever a peak time is found, the following transformations should be computed:

- Average Absolute Acceleration Magnitude Variation, $AAMV = \sum_{t=is}^{ie} \frac{|a_{t+1}-a_t|}{N}$, with is being the impact start, ie the impact end, and N the number of samples in the interval.
- Impact Duration Index, $IDI = impact\ end - impact\ start$. Alternatively, it could be computed as the number of samples.
- Maximum Peak Index, $MPI = max_{t \in [is,ie]}(a_t)$.
- Minimum Valley Index, $MVI = min_{t \in [is-500,ie]}(a_t)$.
- Peak Duration Index, $PDI = peak\ end - peak\ start$, with peak start defined as the time of the last magnitude sample below $th_{PDI} = 1.8 \times g$ occurred before peak time, and peak end is defined as the time of the first magnitude sample below $th_{PDI} = 1.8 \times g$ occurred after peak time.
- Activity Ratio Index, ARI, measuring the activity level in an interval of 700 ms centered at the middle time between impact start and impact end. The activity level is calculated as the ratio between the number of samples not in $[th_{ARIlow} 0.85 \times g, th_{ARIIhigh} = 1.3 \times g]$ and the total number of samples in the 700 ms interval.
- Free Fall Index, FFI, computed as follows. Firstly, search for an acceleration sample below $th_{FFI} = 0.8 \times g$ occurring up to 200 ms before peak time; if found, the sample time represents the end of the interval, otherwise the end of the interval is set 200 ms before peak time. Secondly, the start of the interval is simply set to 200 ms before its end. FFI is defined as the average acceleration magnitude evaluated within the interval.
- Step Count Index, SCI, measured as the number of peaks in the interval $[peak\ time - 2200, peak\ time]$. SCI is the step count evaluated 2200 ms before peak time. The number of valleys are counted, defining a valley as a region with acceleration magnitude below $th_{SCIlow} = 1 \times g$ for at least 80 ms, followed by a magnitude higher than $th_{SCIhigh} 1.6 \times g$ during the next 200 ms. Some ideas on computing the time between peaks [?] were used when implementing this feature.

Evaluating this approach was proposed as follows. The time series of acceleration magnitude values are analyzed searching for peaks that marks where a fall event candidate appears. When it happens to occur, the *impact end* and the *impact start* are determined, and thus the remaining features. As long as this fall events are detected when walking or running, for instance, a Neural Network (NN) model is obtained to classify the set of features extracted.

In order to train the NN, the authors made use of an Activities of Daily Living (ADL) and FD dataset, where each file contains a Time Series of 3DACC values corresponding to an activity or to a fall event. Therefore, each dataset including a fall event or a similar activity -for instance, running can perform similarly to falling- will generate a set of transformation values. Thus, for a dataset file we will detect something similar to a falling, producing a row of the transformations computed for each of the detected events within the file. If nothing is detected within the file, no row is produced. With this strategy, the Abbate et al obtained the training and testing dataset to learn the NN.

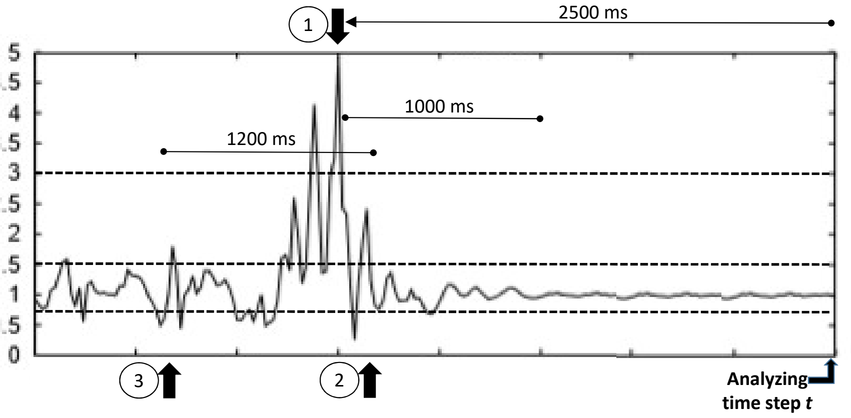

Fig. 1. Evolution of the magnitude of the acceleration -y-axis, extracted from [**?**]. Analyzing at time stamp ***t***, the three conditions described in the text must be found in order to detect a fall. Graph elaborated from [**?**].

2.1 The modifications on the algorithm

As stated in [**?**,**?**], the solutions to this type of problems must be ergonomic: the users must feel comfortable using them. We considered that placing a device on the waist is not comfortable, for instance, it is not valid for women using dresses. When working with elderly people, this issue is of main relevance. Therefore, in this study, we placed the wearable device on the wrist. This is not a simple change: the vast majority of the literature reports solutions for FD using waist based solutions. Moreover, according to [**?**,**?**] the calculations should be performed on the smartwatches to extend the battery life by reducing the communications. Therefore, these calculations should be kept as simple as possible.

A second modification is focused on the training of the NN. The original strategy for the generation of the training and testing dataset produced a highly imbalanced dataset: up to 81% of the obtained samples belong to the class FD, while the remaining belong to the different ADL similar to a fall event.

To solve this problem a normalization stage is applied to the generated imbalanced dataset, followed by a SMOTE balancing stage [**?**]. This balancing stage will produce a 60%(FALL)-40%(no FALL) dataset, which would allow avoiding the over-fitting of the NN models. As usual, there is a compromise between the balancing of the dataset and the synthetic data samples introduced in the dataset.

3 Experiments and results

An ADL and FD dataset is needed to evaluate the adaptation, so it contains time series sample from ADL and for falls. This research made use of the UMA-

FALL dataset [?] among the publicly available datasets. This dataset includes data for several participants carrying on with different activities and performing forward, backward and lateral falls. Actually, these falls are not real falls -demonstrative videos have been also published-, but they can represent the initial step for evaluating the adapted solution problem. Interestingly, this dataset includes multiple sensors; therefore, the researcher can evaluate the approach using sensors placed on different parts of the body.

The thresholds used in this study are exactly the same as those mentioned in the original paper. All the code was implemented in R[?] and caret. The parameters for SMOTE were perc.over set to 300 and perc.under set to 200 -that is, 3 minority class samples are generated per original sample while keeping 2 samples from the majority class-. These parameters produce a balanced dataset that moves from a distribution of 47 samples from the minority class and 200 from the majority class to a 188 minority class versus 282 majority class (40%/60% of balance). To obtain the parameters for the NN a grid search was performed [?,?,?]; the final values were p_size set to 11, p_decay set to 10^{-6} and maximum number of iterations 1000.

Both 5x2 cross validation (cv) and 10-fold cv were performed to analyze the robustness of the solution. The latter cv would allow us to compare with existing solutions, while the former shows the performance of the system with an increase in the number of unseen samples. The results are shown in Table ?? and Table ?? for 10-fold cv and 5x2 cv, respectively. The boxplots for the statistical measurements Accuracy, Kappa factor, Sensitivity and Specificity are shown in Fig. ??.

Fold	Error %	Acc	Ka	Sens	Spec	TP	FP	FN	TN
1	0.0426	0.9574	0.9117	0.9474	0.9643	18	1	1	27
2	0.0625	0.9375	0.8681	0.9444	0.9333	17	2	1	28
3	0.0213	0.9787	0.9562	0.9500	1.0000	19	0	1	27
4	0.0417	0.9583	0.9144	0.9048	1.0000	19	0	2	27
5	0.0851	0.9149	0.8203	0.9412	0.9000	16	3	1	27
6	0.0217	0.9783	0.9539	1.0000	0.9655	17	1	0	28
7	0.0638	0.9362	0.8664	0.9444	0.9310	17	2	1	27
8	0.0426	0.9574	0.9131	0.9048	1.0000	19	0	2	26
9	0.0870	0.9130	0.8175	0.8889	0.9286	16	2	2	26
10	0.1489	0.8511	0.6934	0.8000	0.8889	16	3	4	24
Mean	0.0617	0.9383	0.8715	0.9226	0.9512				
Median	0.0525	0.9475	0.8899	0.9428	0.9488				
Std	0.0382	0.0382	0.0792	0.0532	0.0412				

Table 1. 10 fold cv results. From left to right, the columns stand for the fold number, the classification error, the accuracy, the Kappa factor, the sensitivity, the specificity, and the True Positive, False Positive, False Negative and True Negative results.

Fold	Error %	Acc	Ka	Sens	Spec	TP	FP	FN	TN
1	0.0723	0.9277	0.8511	0.8812	0.9627	89	5	12	129
2	0.0723	0.9277	0.8511	0.8812	0.9627	89	5	12	129
3	0.0766	0.9234	0.8421	0.8800	0.9556	88	6	12	129
4	0.0766	0.9234	0.8421	0.8800	0.9556	88	6	12	129
5	0.0851	0.9149	0.8252	0.8627	0.9549	88	6	14	127
6	0.0426	0.9574	0.9113	0.9468	0.9645	89	5	5	136
7	0.0638	0.9362	0.8673	0.9158	0.9500	87	7	8	133
8	0.0809	0.9191	0.8348	0.8571	0.9692	90	4	15	126
9	0.07234	0.9277	0.8496	0.9053	0.9429	86	8	9	132
10	0.0340	0.9660	0.9288	0.9674	0.9650	89	5	3	138
Mean	0.0677	0.9323	0.8603	0.89778	0.9583				
Median	0.0723	0.9277	0.8503	0.8812	0.9591				
Std	0.0166	0.0166	0.0336	0.0360	0.0080				

Table 2. 5x2 cv results. From left to right, the columns stand for the fold number, the classification error, the accuracy, the Kappa factor, the sensitivity, the specificity, and the True Positive, False Positive, False Negative and True Negative results.

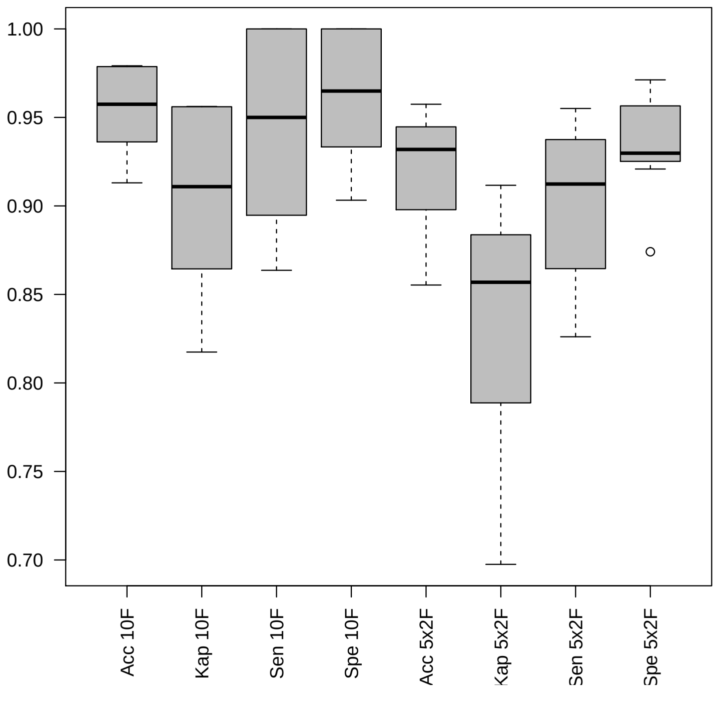

Fig. 2. Boxplot for the different measurements -Accuracy, Kappa, Sensitivity and Specificity-, both for 10 fold cv (four boxplots to the left) and 5x2 cv (four boxplots to the right).

3.1 Discussion on the results

As seen from the previous figures, the classification results after performing the SMOTE are really impressive, with very reduced miss classified samples. As expected, the 10-fold cv results are a bit better than those depicted for the 5x2 cv; nevertheless, the robustness of the method seems pretty good.

After these results, can the problem of FD be considered solved after several minor changes? Answering this question needs discussing some other topics: i) the nature of the dataset, ii) the basis of the method, and iii) the deployment issues and the computational requirements.

The UMA-Fall dataset used in this study [?] was generated with young participants using a very deterministic protocol of activities. The falls were performed with the participants standing still and letting them fall in the forward/backward/lateral direction. Consequently, there could be severe differences between the movements and the data gathered from a real unexpected fall. For instance, after the fall, the participant kept quiet: this is not normal when a person falls unless he/she faints. There are more publicly available datasets, the majority of these datasets have been gathered with healthy volunteers [?,?]. However, a real-world fall and activity of daily living dataset is published in [?], where a comparison of the different methods published so far is also included. Therefore, the method described in this study needs to be validated with more datasets, more specifically, with data from real fall events.

Concerning with the basis of the Abbate et al method, the number of predefined thresholds represents its main drawback. These thresholds have been set for young participants: they might not be valid for a different population. Even if the thresholds are valid, perhaps the classification models must be specific for groups of people according to their movement characteristics [?]. Certainly, there are clear differences in the walking between a young participant and an elder person, even between two elder persons. For instance, using crutches means totally different dynamics. All of these issues must be analyzed in order to validate the approach. Determining the thresholds sets needs further research, trying to find general thresholds -if any-; alternatively, clustering the population according to a given criteria and then finding specific thresholds -and classification models- for each group.

Besides, the eHealth and wearable applications deployment issues have been study in the literature [?]. According to the published results, there is a trade off between the mobile computation and the communication acts to extend the battery charge as long as possible. Consequently, it has been found that moving all the preprocessing and modelling issues to the mobile part could be advantageous provided the computational complexity of the solution is kept low. The consequences of these findings shall be reflected in the transformations and in the models, reducing complex floating point operations as much as possible [?].

4 Conclusions

In this study a fall detection method using a wearable device placed on the wrist is described; the classification method was originally proposed to use a waist located accelerometer. This original approach has been adapted with minor enhancements in the computation of some features; the idea underneath is to detect peaks; for each detected peak a set of 8 features are computed, generating a sample. This sample is labelled according to what has happened -either a fall or the corresponding activity-. With the available samples, the classifying feed-forward Neural Network model can be learned. Nevertheless, this solution clearly generates imbalanced data, which was not considered. In this study, SMOTE was used to balance the training/testing dataset.

The good performance of the method on the UMA-Fall dataset shows this method can represent the basis for a good FD method using wrist-based wearables. However, there are still work to be done, basically, coping with datasets including real falls time series. Also, new solutions need to be proposed to tackle with the thresholds tuning. Furthermore, different solutions for the classification models are needed in order to reduce the floating point operations, so the battery charging cycles could be elongated.

Acknowledgments

This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2014-56967-R and MINECO-TIN2017-84804-R, and FC-15-GRUPIN14-073 (Regional Ministry of Principality of Asturias).