

Peak detection enhancement in autonomous wearable Fall Detection*

Mario Villar¹ and Jose R. Villar²

¹ University of Granada, Granada, Spain mario.villarsanz@gmail.com

² University of Oviedo, Computer Science Department,
Faculty of Geology, Oviedo, Spain
villarjose@uniovi.es

Abstract. Fall Detection (FD) has drawn the attention of the research community for several years. A possible solution relies on on-wrist wearable devices including tri-axial accelerometers performing FD autonomously. This type of approaches makes use of an event detection stage followed by some pre-processing and a final classification stage. The event detection stage is basically performed using thresholds or a combination of thresholds and finite state machines. In this research, a novel event detection is proposed avoiding the use of user predefined thresholds; this fact represents the main contribution of this study. It is worth noticing that avoiding the use of thresholds make solutions more general and easy to deploy. Moreover, a new set of features are extracted from a time window whenever a peak is detected, classifying it with a Neural Network. The proposal is evaluated using the UMA Fall, one of the publicly available simulated fall detection data sets.

Results show the improvements in the event detection using the new proposal, outperforming the base line method; however, the classification stage still needs improvement. Future work includes introducing a finite state machine in the event detection method, adding extra features and a pre-classification of the post-peak interval and a better training configuration of the Neural Networks.

Keywords: Fall Detection, Event Detection, Classification, Wearable Devices.

1 Introduction

The mean age of the Europe's population is rising, which means the society needs to solve some challenges to allow a healthy aging; the detection of fall events or Fall Detection (FD) is among the challenges to solve. Interested readers can find complete reviews on FD in [5,4]. There is a wide spread of studies concerning this topic, such as analyzing video recordings from cameras [21] or using wearable

* This research has been funded by the Spanish Ministry of Science and Innovation, under project MINECO-TIN2017-84804-R, and by the Grant FC-GRUPIN-IDI/2018/000226 project from the Asturias Regional Government.

devices (WD) and the data from the sensors in the detection [12]. The ubiquity of WD leads, in senior houses, to a reduce the effort of the carers to check the inhabitants and also to study their level of activity during each day. Besides, autonomous on-wrist wearable devices, such as smart-watches, including FD might play a crucial role in helping the elder to continue living by their own. These devices can be easily worn because the population usually carry watches or bracelets; because some smart devices are programmable, deploying intelligent services on them is feasible. In this research, we focus on smart-watches with built-in tri-axial accelerometers (3DACC), which is by far the most chosen option in FD with wearables [22,10,20,2,11].

Thresholds have been proposed as the elementary decision system [2,11,13,8,9]; in all these cases, the TS values are analyzed until it surpasses a predefined combination of thresholds or the features extracted from the current sliding window becomes out of the predefined range. Machine Learning is used in the majority of the wearable-based FD to learn the patterns related with falls and with Activities of Daily Living (ADL). Modelling techniques such as Support Vector Machines (SVM), K-Nearest Neighbours, Neural Networks or Decision Trees have been widely used. For instance, SVM were used to classify features extracted from sliding windows [22,20]. A comparison of different classification algorithms has been presented in [10].

There are also studies concerned with the dynamics in a fall event [1,7]. The former study proposed the use of these dynamics as the basis of the FD algorithm [1], with moderate computational constraints but a high number of thresholds to tune. This solution is appealing when developing solutions to run in smart-watches due to their lack of computational power. The proposal of Abbate et al has been modified in a series of papers [14,15,18] to adapt the sensor placement on a wrist. We refer to this event detection as *on-wrist Abbate*. This sensor location introduces changes in the peak detection, forcing to introduce an over-sampling data balancing stage (using the Synthetic Minority Over-sampling Technique - SMOTE) and modifying the feed-forward Neural Network learning process.

The main contribution in this study consist of a new event detection mechanism to detect the high intensity fall events, that is, those that arise when the user stands up and falls either while walking, standing still, running, etc. The mechanism is based on the partial maximum peak detection method [16], where the threshold to detect the peaks is automatically determined for each user. Interestingly, this new event detection makes use of no user predefined threshold, which represents a step ahead in the event detection mechanisms in the literature. We refer to this event detection mechanism as *MAX-PEAK*. Furthermore, a set of transformations will be calculated for the windows surrounding the fall event candidates, and a feed forward Neural Network classifier is used to label the instances.

The structure of the paper is as follows. The next section deals with the description of the on-wrist Abbate event detection method and their transformations. Section 3 details the MAX-PEAK and FSM-MAX-PEAK, together with

the transformations that are proposed to compute and the modelling method. Section 4 describes the UMA Fall data set used and the experimentation. Section 5 shows and discusses the obtained results. Finally, conclusions are drawn.

2 The baseline event detection

As propose in [1,15], the on-wrist Abbate is a simple finite state machine (FSM) - see Figure 1 -. The data gathered from a 3DACC located on the wrist is processed using a sliding window. A peak detection algorithm is executed based on a pre-defined threshold th_1 . When a peak is found, the sliding window data is analyzed in order to extract several features which finally classify the peak as either FALL or NOT_FALL.

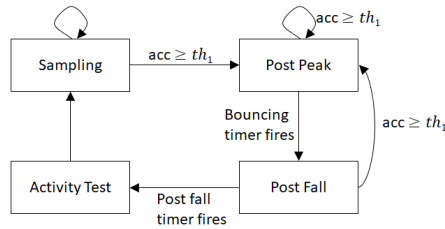


Fig. 1. The on-wrist Abbate FSM. Whenever the acceleration value is higher than a predefined threshold, the state changes to Post_Peak. If no more peaks are detected, the bouncing timer fires and the state moves to Post_Fall, which is supposed to be a calm period without any other peak. Once this second timer fires, the sliding window, still located around the last detected peak, is used to compute several transformations and to classify the sample. Finally, the FSM returns to the initial state.

The Activity Test state computes several transformations in the so called peak-window, which is determined as follows. Let's assume that the gravity is $g = 9.8m/s$. The magnitude of the acceleration at time t is $a_t = \sqrt{a_{tx}^2 + a_{ty}^2 + a_{tz}^2}$ (a_{tx} , a_{ty} and a_{tz} are the acceleration components in each of the axis). A peak occurs at a *peak time* (pt) whenever a_t is higher than $th_1 = 3 \times g$ and there is no other acceleration value above that threshold in the period $(t - 2500ms; t]$ (no other a_t value higher than th_1). The **impact end** ie denotes the end of the fall event: it is the last time moment for which the a_t value is higher than $th_2 = 1.5 \times g$. Finally, the **impact start** (is) denotes the starting time point of the fall event, computed as the starting time of a sequence where $a_t \leq th_3$ ($th_3 = 0.8 \times g$) is followed by $a_t \geq th_2$. The *impact start* must belong to the interval $[ie - 1200 ms, peak time]$. 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 fall-like peak is found, the following transformations should be computed:

AAMV Average Absolute Acceleration Magnitude Variation computed as $AAMV = \sum_{t=is}^{ie-1} |a_{t+1} - a_t| / N$, with N the number of samples in the interval.

IDI Impact Duration Index $IDI = impact\ end - impact\ start$.

MPI Maximum Peak Index $MPI = \max_{t \in [is, ie]}(a_t)$.

MVI Minimum Valley Index $MVI = \min_{t \in [is-500, ie]}(a_t)$.

PDI Peak Duration Index $PDI = peak\ end - peak\ start$, being *peak start* the time of the last magnitude sample below $th_{PDI} = 1.8 \times g$ occurred before pt , and *peak end* the time of the first magnitude sample below $th_{PDI} = 1.8 \times g$ occurred after pt .

ARI Activity Ratio Index, calculated as the ratio between the number of samples that are 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 centered in $(is + ie)/2$.

FFI Free Fall Index, the average acceleration magnitude in the interval $[t_{FFI}, pt]$. t_{FFI} is the time between the first acceleration magnitude below $th_{FFI} = 0.8 \times g$ occurring up to 200 ms before pt ; if not found, it is set to $pt - 200ms$.

SCI Step Count Index, measured as the number of peaks in the interval $[pt - 2200, pt]$.

The on-wrist Abbate FSM is a very challenging event detection that has been successfully used in several studies [14,17,18]. However, it has several drawbacks: the high number of thresholds and the difficulty of the improvements in the feature set. Concerning the high number of thresholds, the peak detection threshold is the main one and can be easily tuned. However, there are many more, some of which are acceleration values and others are time intervals that need to be set. It would be desirable to obtain an event detection and a feature set of transformations free of thresholds: that is the main aim of this research.

3 A new fall detection approach

3.1 The event detection stage

For the purpose of detecting peaks in the 3DACC magnitude, the first stage is to smooth the signal using a sliding window sized $\frac{1}{4}FREQ$, with $FREQ$ being the sampling frequency. Afterwards, we apply the S_1 transformation proposed in [16]. For the current problem, the S_4 and S_5 were too complex for a smart-watch and need too wide windows of data in order to estimate the entropy. From the remaining transformations, we chose S_1 because its simplicity and similar performance among all of them. The Eq. 1 defines the calculation of S_1 , where k is the predefined number of samples and t is the current sample timestamp. It is worth noticing that, although we analyze the window $[a_{t-2k-1}, a_t]$ at time t , the peak candidate is a_{t-k} , the center of the interval. The S_1 transformation represents a scaling of the TS, which makes the peak detection easier using a predefined threshold α .

$$S_1(t) = \frac{1}{2} \times \{ \max_{i=t-2k}^{t-k-1} a_i + \max_{i=t-k+1}^t a_i \} \quad (1)$$

The algorithm for detecting peaks is straightforward: a peak occurs in time t if the value S_t is higher than α and is the highest in its $2k$ neighbourhood. In the original report, all the parameters (k, α) were carefully determined for each problem in order to optimize the peak detection.

The use of predefined threshold is just what is trying to be avoided; therefore, in this research we define simple heuristics to automatically determine them. On the one hand, the value of k is defined with the sampling frequency (the number of samples per second). On the second hand, we define the α threshold as follows. We have walking as the reference activity, so the current user u needs to normally walk during a short period. The mean μ_w^u and the standard deviation σ_w^u for this period are calculated. We also compute the values of S_1 for this walking period, calculating its mean (μ_{w,S_1}^u) and standard deviation (σ_{w,S_1}^u). The TS is then normalized with these statistics; the threshold is set to $\alpha = 3\sigma_{w,S_1}^u$, which means (for a normal distribution) that *a high value that is statistically the upper limit for S_1 when walking is a peak candidate*. With these settings the S_1 is automatically set according to the current device and the user performance. From now on, we refer to this solution as MAX-PEAK.

3.2 The new set of transformations

Whenever a high intensity fall occurs there are three main parts: the activity being carried ordinarily before the fall event, the fall itself that we identify as a peak and what happens next. Because there are no public data set of real falls for healthy participants, we are not able to say accurately what happens after a fall: we can make the hypothesis that what happens after a fall is a period of relative calm, without special activity, perhaps some erratic movements of the hands. Therefore, we will divide the $[a_{t-2k-1}, a_t]$ window in two: before $I_B = [a_{t-2k-1}, a_{t-k-1}]$ and after $I_A = [a_{t-k+1}, a_t]$ the peak. For each of these sub-intervals we propose to compute the following transformations:

AAMV Average Absolute Acceleration Magnitude Variation computed as $AAMV = \sum_{t=s}^{e-1} |a_{t+1} - a_t|/N$, with N the number of samples in the interval $[s, e]$.

E Energy of the Acceleration Magnitude $E = \sum_{t=s}^e a_t^2/N$

Mean Mean Activity the mean of the acceleration magnitude in the interval $[s, e]$.

SD Standard Deviation of the acceleration magnitude in the interval $[s, e]$.

Therefore, we have 4 transformations for each of the two intervals (a total of 8 transformations); none of which relies on thresholds of any kind. All of these transformations are well known in the context of Human Activity Recognition and Fall Detection.

4 Experimental design

The publicly available simulated falls UMA Fall data set [3] is used in this study. This data set includes several activities, transitions and simulated falls

regarding up to 17 participants. There is no fixed number of repetitions of each activity or simulated fall. Each participant used several 3DACC, specially one on a wrist; the sampling frequency was 20 Hz. Altogether, 208 TS are simulated falls, belonging to lateral, forward or backward falls, out of the 531 TS that are available in this data set.

The experimentation is divided in two main parts: the first part is devoted to compare the event detection methods, while the second one aims to evaluate the fall detection algorithm using the feature subset proposed in the previous section.

The comparison of the event detection evaluates both methods (the on-wrist Abbate and the MAX-PEAK) for each participant in the UMA Fall data set. We use the well-known counters True Positive -TP-, True Negative -TN-, False Positive -FP- and False Negative -FN- to evaluate the performance of the event detection methods. These counters are updated according to whether the peaks are detected or not in each TS. Additionally, for every peak the 4 features for each k-length sub-interval are calculated and stored for later used. For now on, this second data set is referred as 4x2TRNS.

For the second part of the experimentation, the 4x2TRNS data set is scaled to the interval $[0.0, 1.0]$. Then, this scaled data set is used to train and test a two classes feed forward Neural Network (NN) (using only the non-fall data) and also a two classes feed forward NN for each type of fall (lateral L-NN, forward F-NN and backward fall B-NN). The train part includes all the instances but those generated for the current participant u ; these instances are used for testing. The sensitivity and specificity of the results for all the participants will be used to measure the performance of the method. For each of the classifiers the best parameter subset will be found using 300 iterations.

5 Obtained results and discussion

Results obtained from the first part of the experimentation, regarding with the comparison of the two methods for detecting falls, are shown in Table 1. As mentioned before, neither the number of TS nor the number of falls for each participant are the same, which in practice will mean that the FP and FN have to be examined in relative terms.

From the very first peek, it can be clearly seen that the MAX-PEAK method is able to detect much more falls (the number of FN is almost naught). Actually, 29.774% of the fall TS result in FN with on-wrist Abbate while MAX-PEAK only lacks in 0.672% FN. Something to emphasise is that standard deviation of the distribution of FN referred to the participants is much more bigger in the case of on-wrist Abbate than for the MAX-PEAK method (0.3351 against 0.0190). This difference highlights that not only on-wrist Abbate suffers from a higher FN rate but it also does it more irregularly.

On the other hand, when talking about FP there is also a clear difference between both methods and, on the contrary, MAX-PEAK has a higher FP rate compared to on-wrist Abbate. Indeed, the MAX-PEAK has an overall ratio of

76.480% of FP whereas on-wrist Abbate’s ratios is just 9.310%. This is due to the fact that Abbate’s method includes a finite state machine which allows filtering the peaks that do not belong to a real fall. For instance, in walking TS there some strong peaks can be found (see Figure 2), but they are usually very close to each other: a finite state machine surely will help to remove those peaks that cannot logically be a fall (for example those pairs of peaks that are closer than two seconds as it is physically impossible to fall two times in such a short period of time). The implementation of a finite state machine for MAX-PEAK is left for further work and research.

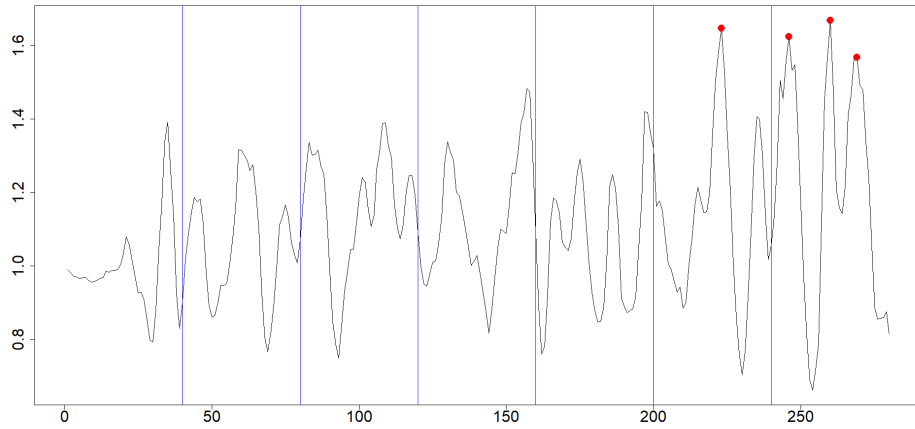


Fig. 2. One of the walking TS graph for participant 1. The red points are the detected peaks, while the blue vertical lines splits in 2-second intervals.

The results obtained from classifying the extracted 4x2TRN data set and the 4 models (NN, L-NN, F-NN and B-NN) are shown in Table 2. The values NA in that Table are due to the fact that several participants did not perform the corresponding type of simulated fall. For instance, participant 5 did not simulate lateral falls, while participant 7 did not simulate any type of fall. The 4x2TRN data set is highly unbalanced (more than 10 Not Fall instances for each Fall one); therefore, a balancing method was used. More specifically, the SMOTE method [6] was used, proposing 500 samples from each class. Nevertheless, the results are highly unsatisfactory as long as the Specificity is negligible in all the cases. Seems that the models are almost always proposing falls for each peak.

Clearly, this part of the proposal needs further attention and improvement. Firstly, the 4 features might not be representative, perhaps it would be more interesting to introduce more features (such as the Signal Magnitude Area [19]). Moreover, it might be interesting to label the post-peak interval with an activity level (high, medium or low) and use this label as an input to the final classifier. Additionally, mainly if the number of transformations gets increased, Principal Component Analysis should be performed first to select the most interesting and

Table 1. Event detection results for each participant.

Pid	on-wrist Abbate				MAX-PEAK			
	TN	FP	FN	TP	TN	FP	FN	TP
1	16	2	5	15	7	11	0	20
2	15	3	2	10	7	11	0	12
3	17	2	2	16	5	14	0	18
4	18	3	7	10	4	17	1	16
5	15	0	0	6	5	10	0	6
6	4	0	2	4	0	4	0	6
7	20	2	0	0	2	20	0	0
8	16	3	0	0	3	16	0	0
9	16	2	2	16	5	13	0	18
10	19	2	0	0	7	14	0	0
11	19	0	1	0	4	15	0	1
12	22	1	9	0	0	23	0	9
13	7	0	5	7	4	3	0	12
14	5	0	0	6	1	4	0	6
15	9	1	8	3	3	7	0	11
16	56	8	5	51	8	56	0	56
17	12	6	8	10	3	15	1	17
Total	286	35	56	154	68	253	2	208

representative axis. However, we haven't introduced all these issues in this study because we need to introduce the finite state machine in the event detection first. All of the proposals are left to future work.

6 Conclusions

This research analyzes an improvement in the peak detection when studying fall detection systems. More specifically, the developed method for detecting peaks amplifies the signal and uses the statistics values to automatically set the firing threshold, adapted to the current user. To our knowledge, this is the first study that proposes an event detection method that automatically adapts to the user behaviour. The results from the experimentation show that the peak detection clearly outperforms the base line method, but suffers of a high percentage of peaks from ADLs. Nevertheless, the automatically threshold set up works extremely well in adapting to each participant. The modeling part still needs more refinements and improvements, which are left to future work.

Future work includes developing a finite state machine for the peak detection that filters the successive peaks within two or three seconds and evaluating with several public FD data sets. Moreover, several transformations should extend the 4x2TRN data set for each interval, Principal Component Analysis will be applied before classifying and also a new stage to label the post-peak interval with the level of activity. Introducing Autoencoders and Deep Learning is also part of future work.

Table 2. Results for the classification task: Sens and Spec stand for Sensitivity and Specificity. The NA stands for Not Available: for these participants there were no simulated falls; therefore, both the TP and FN have a value of 0 and, hence, the Sensitivity is not computed.

	NN		L-NN		F-NN		B-NN	
	Sens	Spec	Sens	Spec	Sens	Spec	Sens	Spec
1	1.0000	0.2710	1.0000	0.0935	1.0000	0.1402	1.0000	0.1308
2	1.0000	0.4343	1.0000	0.0859	1.0000	0.3939	1.0000	0.0657
3	1.0000	0.3571	1.0000	0.0476	1.0000	0.0655	1.0000	0.1548
4	1.0000	0.4688	1.0000	0.0573	1.0000	0.1042	1.0000	0.2552
5	1.0000	0.4386	NA	0.2632	NA	0.4211	1.0000	0.2895
6	1.0000	0.2308	1.0000	0.0000	1.0000	0.3077	1.0000	0.0769
7	NA	0.4388	NA	0.1939	NA	0.1429	NA	0.0612
8	NA	0.5223	NA	0.1401	NA	0.0382	NA	0.2102
9	1.0000	0.4670	1.0000	0.3208	1.0000	0.1934	1.0000	0.0660
10	NA	0.3505	NA	0.1959	NA	0.2887	NA	0.1237
11	1.0000	0.3597	NA	0.4029	NA	0.0935	NA	0.2014
12	1.0000	0.4171	1.0000	0.1991	1.0000	0.1137	1.0000	0.1659
13	1.0000	0.5970	1.0000	0.7761	1.0000	0.5224	1.0000	0.4179
14	1.0000	0.1250	1.0000	0.3750	1.0000	0.2500	1.0000	0.1250
15	1.0000	0.4731	1.0000	0.2688	1.0000	0.3118	1.0000	0.3548
16	1.0000	0.3888	1.0000	0.1367	1.0000	0.0148	1.0000	0.1779
17	1.0000	0.2756	1.0000	0.0321	1.0000	0.0577	1.0000	0.0962

References

1. Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., AlessioVecchio: A smartphone-based fall detection system. *Pervasive and Mobile Computing* **8**(6), 883–899 (Dec 2012)
2. Bourke, A., O’Brien, J., Lyons, G.: Evaluation of a threshold-based triaxial accelerometer fall detection algorithm. *Gait and Posture* **26**, 194–199 (2007)
3. Casilari, E., Santoyo-Ramón, J.A., Cano-García, J.M.: Umafall: A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science* **110**(Supplement C), 32 – 39 (2017). <https://doi.org/doi:10.1016/j.procs.2017.06.110>
4. Casilari-Pérez, E., García-Lagos, F.: A comprehensive study on the use of artificial neural networks in wearable fall detection systems. *Expert Systems with Applications* **138** (2019). <https://doi.org/doi:10.1016/j.eswa.2019.07.028>
5. Chaudhuri, S., Thompson, H., Demiris, G.: Fall detection devices and their use with older adults. *Journal of Geriatric Physical Therapy* **37**, 178–196 (2014)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* pp. 321–357 (2002)
7. Delahoz, Y.S., Labrador, M.A.: Survey on fall detection and fall prevention using wearable and external sensors. *Sensors* **14**(10), 19806–19842 (2014). <https://doi.org/doi:10.3390/s141019806>
8. Fang, Y.C., Dzeng, R.J.: A smartphone-based detection of fall portents for construction workers. *Procedia Eng.* **85**, 147–156 (2014)

9. Fang, Y.C., Dzung, R.J.: Accelerometer-based fall-portent detection algorithm for construction tiling operation. *Autom. Constr.* **84**, 214–230 (2017)
10. Hakim, A., Huq, M.S., Shanta, S., Ibrahim, B.: Smartphone based data mining for fall detection: Analysis and design. *Procedia Computer Science* **105**, 46–51 (2017). <https://doi.org/doi:10.1016/j.procs.2017.01.188>
11. Huynh, Q.T., Nguyen, U.D., Irazabal, L.B., Ghassemian, N., Tran, B.Q.: Optimization of an acc. and gyro.-based fall det. algorithm. *Journal of Sensors* (2015)
12. Igual, R., Medrano, C., Plaza, I.: Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine* **12**(66) (2013)
13. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., Jämsä, T.: Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait and Posture* **28**, 285–291 (2008)
14. Khojasteh, S.B., Villar, J.R., de la Cal, E., González, V.M., Sedano, J., YAZĞAN, H.R.: Evaluation of a wrist-based wearable fall detection method. In: 13th International Conference on Soft Computing Models in Industrial and Environmental Applications. pp. 377–386 (2018)
15. Khojasteh, S.B., Villar, J.R., Chira, C., González, V.M., de la Cal, E.: Improving fall detection using an on-wrist wearable accelerometer. *Sensors* **accepted for publication**, 1–20 (2018)
16. Palshikar, G.K.: Simple algorithms for peak detection in time-series. Tech. rep., Tata Research Development and Design Centre (2009)
17. Tsinganos, P., Skodras, A.: A smartphone-based fall detection system for the elderly. In: Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (2017)
18. Villar, J.R., de la Cal, E., Fañez, M., González, V.M., Sedano, J.: User-centered fall detection using supervised, on-line learning and transfer learning. *Progress in Artificial Intelligence* **2019**, 1–22 (2019). <https://doi.org/doi:10.1007/s13748-019-00190-2>
19. Villar, J.R., González, S., Sedano, J., Chira, C., Trejo-Gabriel-Galán, J.M.: Improving human activity recognition and its application in early stroke diagnosis. *International Journal of Neural Systems* **25**(4), 1450036–1450055 (2015)
20. Wu, F., Zhao, H., Zhao, Y., Zhong, H.: Development of a wearable-sensor-based fall detection system. *International Journal of Telemedicine and Applications* **2015**, 11 (2015). <https://doi.org/doi:10.1155/2015/576364>
21. Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., Li, Z.: A review on human activity recognition using vision-based method. *Journal of Healthcare Engineering* **2017** (2017). <https://doi.org/doi:10.1155/2017/3090343>
22. Zhang, T., Wang, J., Xu, L., Liu, P.: Fall detection by wearable sensor and one-class svm algorithm. In: Huang DS., Li K., I.G. (ed.) *Intelligent Computing in Signal Processing and Pattern Recognition, Lecture Notes in Control and Information Systems*, vol. 345, pp. 858–863. Springer Berlin Heidelberg (2006)