A combined smartphone and smartwatch fall detection system

Author1 and Author2
Affiliation 1
address
address
{email,email}@affiliation1

Author3, Author4, Author5, Author6, Author7, Author8 and Author9
Affiliation 2
address
address
{email1, email2, email3, email4, email5, email6, email7}@affiliation2

ABSTRACT

Falls are behind many elderly hospitalizations and can lead to injuries that greatly debilitate old patients. Much of the deployed fall detection systems rely on the user wearing a personal emergence response device, being conscious and at home. The limitation of the existing systems regarding usability and efficiency has yield an overarching research question on whether systems based on consumer mobile devices can be used as automatic fall detectors for seniors.

In this paper, we specifically look into the accuracy of fall detection system based on an off-the-shelf smartwatch and smartphone. This is done by developing a fall detection system consisting on both a smartphone and a smartwatch. To the authors' knowledge, this is the first study putting together both devices in a fall detection system. We have implemented a combination of threshold based and pattern recognition techniques in both devices, with the intent of having the watch to specially contribute to the specificity of the fall detection strategy.

We tested the accuracy of the system through a series of simulated falls and activities of daily living (ADL), resulting on the correct identification of 81% of the falls and 75% of the ADLs and outperforming two other baseline fall detection applications (iFall and Fade). Still, the accuracy of the system could be greatly improved by changing the thresholds of the pattern recognition algorithm used. When it comes to the watch, the algorithm using its sensed data was able to confirm all the tested falls and contribute to the system for distinguishing 25% of ADLs which were not detected using the phone sensor data.

Author Keywords

wearable computing; smartphone; smartwatch; fall detection; health ageing; mobile computing.

INTRODUCTION

Around one-third of elderly fall at least once a year [1]. Elders living alone risk having their fall unnoticed and not being able to stand up by themselves as ask for help. [24] indicates that 47% of elderly fallees were unable to stand-up without assistance in at least one fall. On the other hand, a long waiting time before assistance increases the chances of hospitalization and death [9]. Therefore, an early detection and assistance in case of a fall is important for increasing elderly life expectancy.

Much of the existing fall detection systems rely on some sort of push-button device (in the shape of a pendant or wristband) which the user is supposed to activate after a fall and which will connect to an alarm center through an in-house communication system. Such systems will not trigger the alarm in case the one falling is unconscious or is outside of his home. Those limitations together with the dropping cost of accelerometer hardware have triggered an enormous growth on the research of automatic fall detection systems based on accelerometers [20].

A recent systematic review on automatic fall detection systems built on body-worn sensors [20] gathered 96 publications from 1998 to 2012. From within that list, around 15% have used sensor placements around the thigh, equivalent to a pants front pocket placement, and 8% have tried the wrist. The majority of the studies (66%) targeted the area of the waist/trunk.

While a trunk/waist placement is known for providing accelerometer data that can better model a fall, sensors placements on the trunk tend to generate discomfort to elders and suffer from a high risk of not being used in practice [13]. Indeed, [11] shows that many users of the push button based system have fallen and the alarm was not triggered because they were not wearing the system. Requirements from elderly users drawn on [22, 12] shows that fall detection devices must stigmatize them nor disturb their daily life. In order to accommodate such requirements, trunk/waist based sensors would need first to become cheap and seamlessly integrated (such as into clothing). Therefore, there is a need to look further into the usage of other sensors and placements.

The above mentioned elderly users requirements motivated us to build a system using devices in which the user is likely to incorporate on his daily usage without introducing any burden or generating any sort of stigmatization. We decided to use a system composed by a standard off-the-shelf smartphone (to be worn in the user's pocket, as many users would do) and a smartwatch. Both are consumer devices that would be used also out of the fall detection context and, therefore, would not trigger negative stigmatization.

A smartphone can already be considered as a ubiquitous device and, even if its penetration with a senior audience is still not as high as with youngsters (37% against 71% in developed countries [6]), it is safe to assume that its penetration will increase. This is especially true when we consider that companies, such as Doro [1], are starting to tailor smartphones interfaces to older audiences. Smartwatches are starting to enter strongly on the consumer market after Google launched an Android Operating System (OS) version for those devices and most of the big smartphone manufacturers started producing smartwatches. [13] points that the wrist is the best sensor placement in terms of usability, and if we take into account that the watch is both non-stigmatizing and can offer other uses for its owner, its acceptance is expected to be high.

When it comes to the usage of smartphones and/or watches for fall detection, there has been much more research on algorithms and implementations on phones than watches or wrist-places sensors. As presented in [17] there are more than 50 different scientific publications developing smartphone-based fall detection systems. On the other hand, when it comes to watches and wrist based sensors, this number is reduced to 8 [20]. From those 8, [5] and [23] used a system based on device with a real watch form factor, while others mainly attached sensors to the wrist region.

From within the articles testing wrist placement, both [16] and [13] performed fall tests using both wrist and other placements of sensors together. [13] looked into the sensitivity of a few algorithms for different placements for forward, backward and lateral falls. It did not look into their performance in ADLs nor on how the sensors could complement each other, but it noticed that the detection of the fall impact phase with the wrist data performed poorly for lateral falls.

[16] examined 5 different sensor placements, including the wrist and the thigh. However it mainly observed the data from the different placements as to describe different falls in terms of different sequence of patterns captured by the different sensors in a conjunct. As a consequence of that, there was limited insight on how much a wrist sensor could complement a thigh one or other set-up then the one in the experiment. However, it was able to indicate that the wrist can help identifying sitting situations and slow falls. On top of that, together with [2] and [8], it has shown that merging the sensing data from different placements has led to an increased sensitivity of detection of falls.

That being said, this is the first, to our knowledge, published work specifically looking at an automatic fall detection system composed exclusively of two sensors located on the thigh region (in the front pocket) and on the wrist. As a consequence of that, it is also the first one to use consumer devices for both sensors.

[1] `http://www.doro.no/`

Besides the development of the system per se, this work tries to answer the following research questions:

1. Which level of fall detection can we achieve using a mobile phone together with a smart watch?

2. How the sensor reading from the watch can add up and improve a phone based fall detection system? Can it help reducing false positives?

In the next section we describe the implementation of the system illustrating the choices of algorithms used and how do they fit the system architecture. The system architecture is described there as to illustrate how the constraints and features of the mobile platforms were taken into account and how the user interface (UI) interacts with the system components.

In the Evaluation section, we describe the different test cases, the accuracy of the system per test case, and how it compares with existing systems. Then, we conclude with the analysis of the test results and suggestions of future work for further expanding the findings of this research.

THE FALL DETECTION SYSTEM

Hardware and baseline software

This research is aimed to be applied in regular consumer devices in the format of smartwatches and smartphones. As the android smartphones have incorporated a large array of sensors including accelerometers and gyroscopes, more and more fall detection research is being developed on top of Android [17].[10] has shown that the smartphone accelerometers are as other accelerometers for detecting falls with high accuracy. On top of that, Android is the dominant smartphone OS and it is expected to become the dominant smartwatch platform[2]. Therefore, we decided use both android smartphones and smartwatches as the platform for our system.

Android based smartwatches run an Android OS version specially designed for wearable devices called Android Wear[3]. In general, Android Wear watches are equipped with Bluetooth low energy (BLE) radio for communicating with smartphones without using much battery and with native inertial measurement sensors.

For the smartwatch, we used the LG G Watch R[4] as it runs Android Wear and possess both a BLE radio and a 9-axis motion sensor combining a 3-axis gyroscope, 3-axis accelerometer and 3-axis compass. As for the smartphone, we used the Samsung Galaxy S3[5] which contains a similar 9-axis motion sensor and BLE radio. The Samsung Galaxy S3 runs Android 4.4, which is compatible with the Android Wear requirements, and therefore, capable of communicating with the LG G Watch R.

[2] `http://www.alliedmarketresearch.com/smartwatch-market`

[3] `http://www.android.com/wear/`

[4] `http://www.lg.com/global/gwatch/index.html#specification`

[5] `http://www.samsung.com/global/galaxys3/`

The adoption of Android by the fall detection researchers is very positive as it opens the door to non-research oriented developers to benefit from the research findings. However, as discussed in [17] very few of the researched apps for fall detection have been released to the public, and, as far as our knowledge goes, none of them have been released as open source and published in a publicly accessible repository. For that reason, we introduced one extra non-scientific requirement to our research about releasing the results and artifacts of this research as open source. We drew inspiration from EU's Open Science Vision for research [7] and decided to release the source code, its documentation and a more thoughtful documentation of the tests in ANONYMYZED LINK TO GITHUB REPOSITORY.

Fall Algorithm

The main goal of this research is to build a fall detection system which takes advantage of the motion sensors both on a smartphone and a smartwatch, assess its accuracy and the benefit of integrating the watch as part of the system. Since there were no published open source android fall detection applications, we had first to choose and implement a fall algorithm both on the smartphone and on the watch.

Fall detection strategies may differ in terms of the types of movement they try to identify (as part of the identification of a fall) or the method used to detect them: fixed thresholds, acceleration patterns, fuzzy logic and artificial intelligence (AI) methods. A fall can be modeled as staged event consisting of essentially 5 phases [18] (where the three intermediary phases are highlighted and illustrated through the acceleration amplitude of one of our test cases in Figure 1):

- **Pre-fall phase**: an initial activity of daily living of which the subject is engaged in
- **Free-fall phase**: the free fall movement towards the ground derived from the loose of balance
- **Impact phase**: the impact representing the moment in which the subject touches the surface
- **Post-impact phase**: the instant after the impact when the person lies inactive on the floor due to the shock of the fall
- **Recovery phase**: the subjects effort to stand-up or recover from the fall

For this research, we have implemented both a threshold based and a pattern recognition algorithm in both the watch and the phone. Threshold based algorithms use one or more formulas against a threshold that can represent a phase of a fall (usually both free-fall and impact phases). It is the simplest solution for detecting a fall and it can be used together with some form of pattern recognition in order to be more reliable. Threshold based is the most commonly used method for fall detection [20] and it has been previously implemented and well described in other research articles targeting Android platforms [17, 4]. We followed the lead of those articles and narrowed down our selection to 3 different threshold formulas:

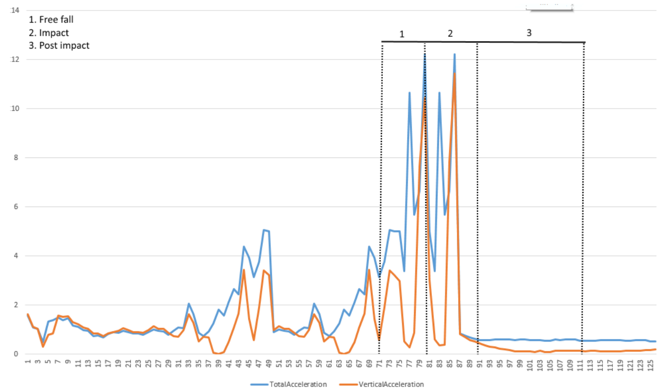

Figure 1: Fall phases represented on acceleration graphs

1. <u>Total vectorial acceleration</u>: This is the same formula used to calculate the Euclidean vector distance, but used for calculating the total acceleration at one point in time regardless of its direction. Due to its simplicity, it is the most used formula for detecting falls [19], but it is rarely used alone. It uses the acceleration values in *x*-, *y*- and *z*- axis.

$$| A_T |= \sqrt{| A_x |^2 + | A_y |^2 + | A_z |^2} \quad (1)$$

2. <u>Fall Index</u>: It is based on the sums of the difference in acceleration in all directions between adjacent data points until the *j*-th previous sample. As the formula relies on the use of adjacent samples, it requires a high sampling frequency for being efficient.

$$FI_i = \sqrt{\sum_{k=x,y,z} \sum_{i-j}^{i} ((A_k)_i - (A_k)_{i-1})^2} \quad (2)$$

3. <u>Absolute vertical acceleration</u>: This formula calculates the total acceleration in vertical direction. Considering that a fall contains a large vertical acceleration there should not be much difference between this value and the value of the total acceleration in a fall.

$$| A_v |=| A_x sin\Theta_z + A_y sin\Theta_y - A_z cos\Theta_y \cdot cos\Theta_z | \quad (3)$$

On the other hand, pattern recognition algorithms use a series of the movement readings patterns and compare with databases or knowledge gathered from training sets. Pattern recognition is often implemented to detect fall stages and differentiate fall events from ADLs. The sensor data collected after a suspicious acceleration threshold can be used to distinguish if a person is lying still after high acceleration (which characterizes a post-fall phase) or if there is continuous movement (which can indicate that he is engaged in other ADL). Therefore, it is able to provide an extra insight in the fall detection on top of a threshold approach. However, pattern recognition algorithms use more memory and are heavier to compute for the device.

The final fall detection strategy implemented on the system has been to have the phone as the main device for the fall detection as its placement, the thigh, is more favorable for identifying falls. The strategy consisted on using three different approaches for detecting the fall in this presented order:

1. Phone Acceleration Threshold (PAT)
2. Phone Pattern Recognition (PPR)
3. Watch Threshold and Pattern Recognition (WTPR)

PAT is based on the monitoring of the acceleration threshold on the phone and is the only approach running continuously as it is the lightest one in terms of processing power and it does not require exchanging data between the devices. The monitoring is based on a sampling of 25 Hz, where the readings are used in formula 1 and 3 and compared with theirs respective thresholds A_{Tt} and A_{vt}. If both accelerations exceed their set thresholds, the algorithm will compare the two acceleration values by calculating the quotient of the division of A_{Tt} over A_{vt} and check it against a threshold: A_{rt}. This is to find out how much of the total acceleration is vertical. Such comparison helps to eliminate false positives due to ADLs involving non-vertical acceleration such as walking or running.

If the three acceleration thresholds are exceeded, the PPR will be executed. Our phone pattern recognition algorithm was based on an initial collection of training sensing data and fall patterns described in the literature. It uses the phone sensed data to identify the free-fall, impact and post-impact phases. It first finds the sample with highest total acceleration (A_{TH}) within a 2 seconds dataset of sensor readings. Then, it checks whether there has been a sudden increase in acceleration prior to it and a sudden decrease after it. The sudden increase is detected when the ratio between A_{TH} and the lowest total acceleration(A_{TL}), in a 0,4 second prior to A_{TH}, is over the free-fall threshold T_{ff}.The sudden increase is calculated on the same fashion as the increase, but considering the lowest total acceleration(A_{TL}) within 0,4 seconds after $A_T H$, and by comparing it a specific impact ratio threshold T_i. If both ratios are over their thresholds, the phases are detected and the algorithm analyzes the average acceleration pattern from the impact for 0,8 seconds. If the acceleration samples during that period are under the post-impact threshold T_{pi}, it indicates that the person is lying fairly still, corresponding to the post-fall phase. The identification of those fall stages in their sequence triggers the algorithm to detect the event as a fall.

If a fall is confirmed by the PPR, the watch data is pulled so that the watch threshold and pattern recognition algorithm is executed. Due to a limitation on the watch for providing reliable orientation data, we based its threshold detections on the Fall Index formula. We use the formula to identify if the difference of acceleration on the impact phase is above a threshold F_{Ti} and if the difference of acceleration during the post-impact phase is under the threshold F_{Tpi}. The watch data is also be used to detect a short resting pattern from the post-impact phase. This is done by checking that for all the samples within 0,8 seconds after the impact, the difference of acceleration to the power of two of each pair is lower than the watch's post-impact threshold T_{pi}.

System Architecture

The architecture of the system was designed so that the phone would be the core of the system and that the system would be expandable towards integrating other external sensors and wearable devices. Since the sensor readings act as the trigger for the algorithms, we opted to use an Event-driven architecture (on top of the EventBus library[6]), where any sensor data publisher can publish data onto the bus, and then in turn, any subscriber can grab the data it needs from the bus. Doing it so allowed the different services to run in parallel and communicate asynchronously as it will be described in the rest of this subsection and represented in the sequence diagram on Figure 2.

Each sensing device, in our case the smartwatch and the smartphone, implements a SensorManager class which is responsible for retrieving the readings from the device and forwarding the data to the event bus. The SensorManager registers with the Android OS Sensor Service specifying the sensors and the sampling frequency it is interested in. Then, upon each reading, the SensorManager packs the data, which initially contained only the raw value and the time stamp, with the information identifying the sensor, its type and its placement on the user body.

In the case of the mobile phone, the SensorManager directly publishes the sensed packets on the bus. For the watch, its SensorManager has to transfer the packet to phone, to them publish it to the bus. The RemoteSensorManager class on the phone is the one responsible for publishing the packets coming from the watch or other Android Wear device into the bus. The SensorManager on the watch is also configurable for sending the data continuously or to just pack and send the data when the phone requests (when both PAT and PPR identify a fall). In this later case, a buffer is maintained in order to ensure that the sensor data is stored for when the phone needs it.

The smartphone software also implements a Service, here refereed as the Fall Algorithm Controller, which subscribes to the Eventbus and retrieves the sensor readings packets published by all SensorManagers. For research and test purposes, multiple fall detection strategies could be implemented as different services subscribed to the bus and process the same data. The Fall Algorithm Controller process collects the data accumulated in a 2 seconds period, having an one second overlap of sensed data between each sequential data collection, and uses the collection as input for the algorithms. The three fall phases our algorithms look at should all occur within 2 seconds.

The accelerometer data is represented in a *x*-, *y*- and *z*- axis relative to the device surface. The Fall Algorithm Controller converts the phone data to a coordinate system relative to the ground using the compass and rotation information. A similar conversion could not be reliably done to the watch accelerometer data due to the fact that the accelerometer and compass readings are not provided by Android Wear with the same sampling timestamp. Such timestamp difference

[6] `https://github.com/greenrobot/EventBus`

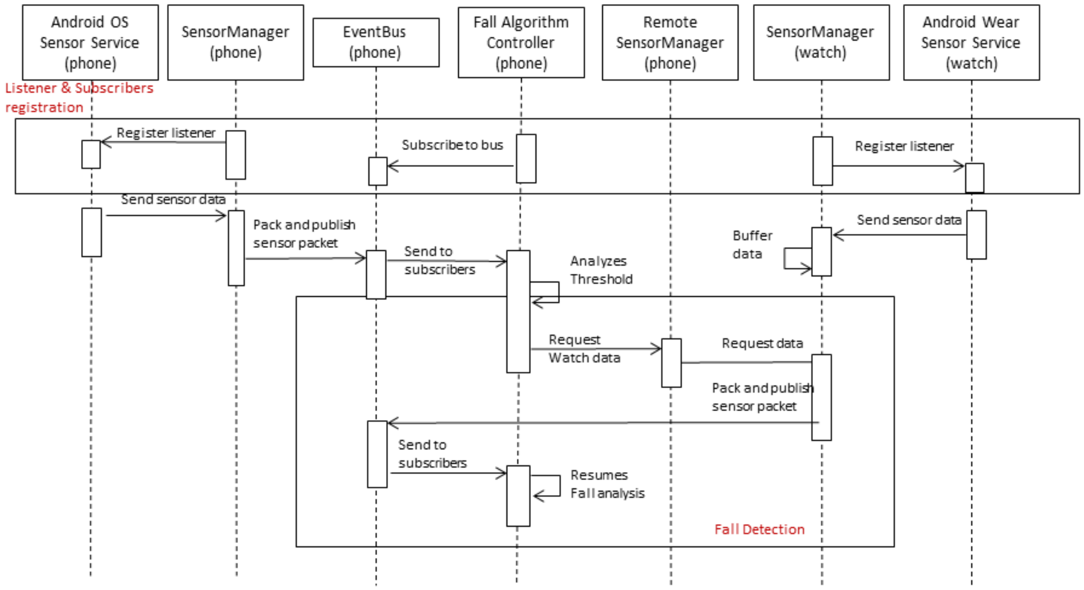

Figure 2: Overall system diagram illustrating the communication between the main components

together with the high degree of movement of the arm and wrist would not allow us to approximate the timestamps of accelerometer and compass without introducing errors. However, as we previously mentioned the algorithm using the watch data does not need to know the orientation of the acceleration towards the ground.

The converted phone data is then processed following the detection strategy presented on the previous sub-section. If the three identification methods converge, we signalize the event as fall and trigger the fall alarm.

User Interface

Similarly to other android fall detection studies mentioned here, and as expected of an automatic fall detector, the user interface was made it simple and without the need frequent interaction. On its first usage, the application starts by showing a short wizard explaining how it works and retrieving the emergency contact information to be used in case of a fall as illustrated in the Figure 3. From then and on, the application will be running on background and it will keep an Android notification visible to signalize it, as explained on the wizard (fig. 4) and visible in its top left of the figure.

The user can re-visit the application, change the emergency contact or even go through the wizard again. However the only event which will require his attention is the detection of a fall. When a fall is detected, the application will play a sound and display a timer button both on the watch as on the phone (Figure 5). If the user press the button, that means that it was a false positive or that he does not need assistance, and therefore no alarm is generated. If the timer expires (the yellow circular border becomes red), the phone will send a text message to the emergency contact and try to call him.

EVALUATION

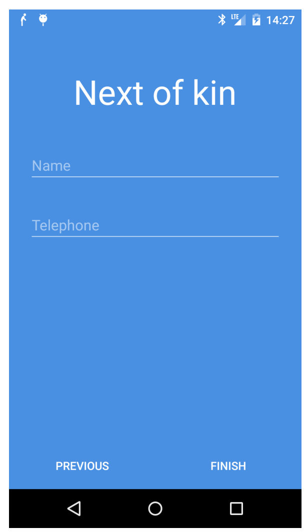

Figure 3: Next of a kin registration

Figure 4: One of the wizard's screens

The goal of this study was to assess the accuracy of the system and to identify the contribution of the watch to the smartphone based fall detection. [19] shows that more than 70 different types of ADLs and more than 40 different types of falls have been tested across different studies. In order to define our set of fall and ADL patterns to be tested, we decided to use a as basis the cases from [13], since it is a widely cited publication and it includes the mostly used test cases reported in the review study [19]. Based on that, our test case consisted the 12 fall patterns and 8 ADL patterns below.

Fall Test Cases:

1. **Normal fall**: faint fall forward with knee bent

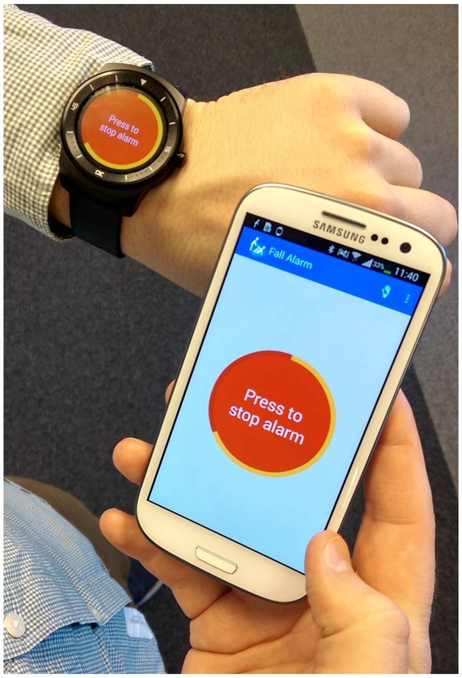

Figure 5: Phone and watch GUI when a fall is detected

2. **Falling backwards**: Faint fall backwards with a round back and knee bent
3. **Step down from platform**: Step down off a platform and fall forward in the process of stepping down
4. **Falling backwards, failing to sit down**: Backward sitting-on-empty on the floor, no arm use, no step back
5. **Falling sideways, left landing on wall**: Side fall to the left landing at the base of a wall
6. **Falling sideways, right landing on wall**: Side fall to the right landing at the base of a wall
7. **Falling from a sitting position**: Falling of a chair: sitting on edge and slipping of
8. **Self tripping**: Walking tripping on owns foot
9. **Falling backwards against wall**: Backward fall at the base of a wall
10. **Falling left**: Faint fall left with knee bent
11. **Falling backwards/left**: Fall backwards and turning to the left side
12. **Falling backwards/right**: Fall backwards and turning to the right side

ADL Test Cases:

1. **Sitting down slowly**
2. **Sitting down quickly**
3. **Walking** (at least 10 meters)
4. **Jogging** (at least 10 meters)
5. **Turning around**: Quickly turn (spin) 360 degrees
6. **Tying shoes**: Crouching (going on the knees) and tying shoes
7. **Elevator**: Riding an elevator from the 5th floor to 0 and back
8. **Stairs**:Going down and up stairs (at least 10 steps each way)

The tests were performed three times each by a 22 years old volunteer of 185 cm and 75 kg instructed to fall as naturally as possible in a 55mm soft mattress placed on top of a hard 35mm martial arts tatami. No additional instructions were given regarding the movements of the arms as there are no published guidelines on the engagement of the subjects‘ arms during fall tests using wrist sensors. However, in order to reduce the bias, we compared the recordings of the test cases performed using the watch with recordings of the same test cases without the watch (which were previously performed to gather training sensor data for the phone algorithm).

The thresholds described in the Table 1 were used in the system in order to assess its accuracy. Those values were based on the analysis of a training dataset generated when testing the system correctness and the values used by articles referred in this work.

PAT	PPR	WTPR
$A_{Tt} = 11m/s^2$	$T_{ff} = 1.5$	$F_{Ti} = 30m/s^2$
$A_{vt} = 9m/s^2$	$T_i = 1.5$	$F_{Tpi} = 20m/s^2$
$A_{rt} = 0.5$	$T_{pi} = 6m/s^2$	$T_{pi} = 170(m/s^2)^2$

Table 1: Thresholds values used during the tests

It was difficult to find a suitable benchmark to compare with our results properly. As described earlier, different studies have used distinct test cases and, on top of that, described the accuracy of theirs systems in different ways. Most studies we have read just describe the total sensitivity and specificity instead of providing those numbers per test case. The difference on test cases together with the lack of details on the results, make it impossible to establish a comparison without a bias.

For that reason, we decided to benchmark our tests against iFall [21] and Fade[7] and to identify the effects of the different algorithms used within our system. iFall is the only publicly available Android fall application we found whose fall strategy is documented, and it has been already tuned and used as benchmark by other researchers [17]. Fade does not have its fall detection strategy published, but it was the one who performed the best among the five top ranked applications on Google's Playstore[8]. The Table 2 summarizes the test results by presenting:

- The accuracy of falls and ADL detection per application for each test case

[7]`http://fade.iter.es/`

[8]`https://play.google.com/store?hl=en`

Test Case	IFall	Fade	Our System	Algorithm
Normal fall	100%	100%	67%	PPR
Falling backwards	100%	67%	67%	PPR
Step down from platform	67%	0%	100%	-
Falling backwards, failing to sit down	0%	100%	67%	PPR
Falling sideways, left landing on wall	33%	0%	67%	PPR
Falling sideways, right landing on wall	33%	67%	100%	-
Falling from a sitting position	67%	33%	67%	PPR
Self tripping	67%	67%	100%	-
Falling backwards against wall	33%	0%	100%	-
Falling left	0%	0%	67%	PPR
Falling backwards/left	0%	100%	67%	PPR
Falling backwards/right	0%	33%	100%	-
Sitting down slowly	100%	100%	100%	PAT/PAT/PAT
Sitting down quickly	100%	67%	67%	PAT/PAT/ -
Walking	100%	100%	67%	PPR/PAT -
Jogging	67%	67%	0%	- / - / -
Turning around	100%	67%	100%	PAT/WTPR/PAT
Tying shoes	100%	100%	67%	PAT/ - /PAT
Elevator	100%	67%	100%	PAT/PAT/PAT
Stairs	100%	0%	100%	WTPR/PAT/PAT
Sensitivity	42%	47%	81%	
Specificity	96%	71%	75%	
Accuracy	63%	57%	78%	

Table 2: Summary of the tests

- The sensitivity, the specificity and accuracy (ratio of correct detection based on both falls and ADLs)
- The identification mechanism within our system which failed to detect a fall case (keeping in mind that PPT, PPR and WTPR are executed sequentially and just upon the condition that the previous algorithm detected the event as a fall)
- The identification mechanism within our system which identified the ADL per test instance

The test cases were recorded both in video and as accelerometer data. They were published at the project's github in order to easily enable others to reproduce them, test different thresholds and strategies or use them as benchmark.

CONCLUSION AND FUTURE WORK

The tests done so far have been useful for providing an initial assessment of the system. The results of the fall cases show that the PAT alone would have been able to detect 100% of the falls, and that WTPR was also capable of positively identifying all the fall events in which it was consulted. On the other hand, the phone acceleration threshold algorithm was responsible for ruling out only 62,5% of possible false positives, where the watch contributed by identifying 8% of the total (25% of the ADLs submitted to the WTPR).

The PPR was able to detect only one ADL wrongly identified by the PAT and it failed detecting falls on 19% of the cases. This, and also the acceleration graphs of the tests, show that the PPR and its thresholds should be revised. As it is, the removal of the PPR from the system could potentially increase the sensitivity to 100% while decreasing the specificity of 4%, but we can not be sure as WTPR could have failed in those test instance. In any case, the tuning of the PPR should be analyzed as to reduce false positives without affecting the fall detection and the next tests should always include the WTPR in order to sheer more light to this matter. In fact, a logical next step would be to always run both WTPR and PPR when PAT is triggered and change the current combination of fall identifications to use fuzzy logic as it would allow us to map the different accuracy of each algorithm to its confidence.

In total, we performed 60 tests. Although this number was enough to assess the accuracy of the system and aspects to be improved, it definitively needs to be expanded. More tests could help us truly investigate the different algorithms performance on a per test case basis and to come up with better thresholds. Most of the published articles we have read (such as [17, 4]) used around 500 test samples to build and test theirs thresholds and strategies. Besides conducting more tests ourselves, we hope that the open publication of the data and the videos illustrating the fall cases from this work can boost the development of a more representative database of falls on smartphones and smartwatches.

Still on the testing, it would be interesting to study the algorithms performance for other typical phone placements such as in a jacket, purse, back pocket or even when the user is talking on the phone or interacting with its keypad. [25] has started this work by finding thresholds based on the total vectorial acceleration resulting on around 80% accuracy.

The future work related to test cases should also include test subjects which are more representative and ideally real falls. Both the usage of young subjects [14] and simulated fall [3] introduce some bias in the results as they lack precision on representing real elder falls. However, capturing real-world falls is very difficult as it needs a large number of subjects running the system for a long time in order to collect enough data [3]. Such real-world trials may be not feasible with the accuracy of the system at the moment, but when the accuracy is improved it would be realistic to consider a wide real world trial due to the large adoption of Android devices and the ease of distributing the app globally through Appstores.

ACKNOWLEDGMENTS

The work behind this study has received funding from the ANONYMIZED FOR SUBMISSION under grant agreement no ANONYMIZED FOR SUBMISSION.

REFERENCES

1. Ageing, W. H. O., and Unit, L. C. *WHO global report on falls prevention in older age*. World Health Organization, 2008.
2. Aziz, O., and Robinovitch, S. N. An analysis of the accuracy of wearable sensors for classifying the causes of falls in humans. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on 19*, 6 (2011), 670–676.
3. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., Zijlstra, W., and Klenk, J. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one 7*, 5 (Jan. 2012), e37062.
4. Dai, J., Bai, X., Yang, Z., Shen, Z., and Xuan, D. PerFallD: A pervasive fall detection system using mobile phones. *Proc. of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops 10* (2010), 292–297.

5. Degen, T., Jaeckel, H., Rufer, M., and Wyss, S. SPEEDY:a fall detector in a wrist watch. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, 184–187.

6. Deloitte. The smartphone generation gap : over-55 ? there s no app for that. Tech. rep., 2014.

7. "European Commission Directorate General for Communications Networks,Content and Technology". Digital science in Horizon 2020. Tech. Rep. March, 2013.

8. Gjoreski, H., Luštrek, M., and Gams, M. Accelerometer placement for posture recognition and fall detection. *Proceedings - 2011 7th International Conference on Intelligent Environments, IE 2011* (2011), 47–54.

9. Gurley, R. J., Lum, N., Sande, M., Lo, B., and Katz, M. H. Persons found in their homes helpless or dead. *New England Journal of Medicine 334*, 26 (1996), 1710–1716.

10. Habib, M., Mohktar, M., Kamaruzzaman, S., Lim, K., Pin, T., and Ibrahim, F. Smartphone-Based Solutions for Fall Detection and Prevention: Challenges and Open Issues. *Sensors 14*, 4 (2014), 7181–7208.

11. Heinbüchner, B., Hautzinger, M., Becker, C., and Pfeiffer, K. Satisfaction and use of personal emergency response systems. *Zeitschrift fur Gerontologie und Geriatrie 43*, 2 (2010), 219–223.

12. Holliday, N. Fall detectors what do users want ? Tech. rep., Health Design & Technology Institute,Coventry University Technology Park, 2012.

13. Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jämsä, T. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture 28*, 2 (Aug. 2008), 285–91.

14. Klenk, J., Becker, C., Lieken, F., Nicolai, S., Maetzler, W., Alt, W., Zijlstra, W., Hausdorff, J., Van Lummel, R., Chiari, L., et al. Comparison of acceleration signals of simulated and real-world backward falls. *Medical engineering & physics 33*, 3 (2011), 368–373.

15. Li, Q., and Stankovic, J. A. Grammar-based, posture- and context-cognitive detection for falls with different activity levels. In *Proceedings of the 2nd Conference on Wireless Health - WH '11* (2011).

16. Luque, R., Casilari, E., Morón, M.-J., and Redondo, G. Comparison and Characterization of Android-Based Fall Detection Systems. *Sensors 14*, 10 (Jan. 2014), 18543–18574.

17. Noury, N., Rumeau, P., Bourke, a. K., ÓLaighin, G., and Lundy, J. E. A proposal for the classification and evaluation of fall detectors. *Irbm 29*, 6 (2008), 340–349.

18. Pannurat, N., Thiemjarus, S., and Nantajeewarawat, E. Automatic fall monitoring: a review. *Sensors 14*, 7 (Jan. 2014), 12900–36.

19. Schwickert, L., Becker, C., Lindemann, U., Maréchal, C., Bourke, a., Chiari, L., Helbostad, J. L., Zijlstra, W., Aminian, K., Todd, C., Bandinelli, S., and Klenk, J. Fall detection with body-worn sensors : a systematic review. *Zeitschrift für Gerontologie und Geriatrie 46*, 8 (Dec. 2013), 706–19.

20. Sposaro, F., and Tyson, G. iFall: An android application for fall monitoring and response. *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009* (2009), 6119–6122.

21. Steele, R., Lo, A., Secombe, C., and Wong, Y. K. Elderly persons' perception and acceptance of using wireless sensor networks to assist healthcare. *International journal of medical informatics 78*, 12 (Dec. 2009), 788–801.

22. Tan, T.-H., Gochoo, M., Chang, C.-S., Wu, C.-T., and Chiang, J. Y. Fall detection for elderly persons using android-based platform. *Energy 2*, 2 (2013), 2.

23. Tinetti, M. E., Liu, W.-L., and Claus, E. B. Predictors and prognosis of inability to get up after falls among elderly persons. *Jama 269*, 1 (1993), 65–70.

24. Viet, V., and Choi, D.-j. Fall Detection with Smart Phone Sensor. *Proceedings of the 3rd International Conference on Internet (ICONI)*, December (2011), 15–19.