Analysis of a hybrid Android system for fall detection

Eduardo Casilari and Miguel A. Oviedo-Jiménez

Abstract**—** Android personal devices have become an interesting and cost-effective technology to deploy wearable Fall Detection Systems. In contrast with other smartphone-based solutions, this paper describes a fall detection architecture that integrates two-Bluetooth enabled devices: a smartwatch and a smartphone. The evaluation of the system under different fall recognition algorithms and mobility patterns indicates that the simultaneous operation of the two devices as fall detectors clearly improves the specificity of the system when compared to the cases where just one device is employed as a fall detector. The performed analysis also encompasses the study of the battery consumption and the performance of the system under constant monitoring in everyday life conditions.

Keywords**—** Tele-healthcare, Telemonitoring, mHealth, Fall Detection Systems, Android, Smartphone, Smartwatch.

I. Introduction

Falls have become a major cause of unintentional injuries among seniors and, consequently, a key concern for national health systems. Diverse studies by the World Health Organization [1], [2] have reported a significant recurrence of falls among the elderly as long as 28%–35% of the population over 64 experience at least one fall annually. Direct health costs due to falls of older adults in USA totaled $34 billion in 2013 [3], and they are projected to rocket in the next years.

The rates of the morbidity and mortality provoked by falls have been proved to be strongly dependent on the speed of the response and the medical first aids after the fall [4]. Thus, during the last decade, many research efforts have been dedicated to deploy cost efficient and reliable Fall Detection Systems (FDS).

Most FDSs in the literature can be categorized into two general groups [5]. On one hand, Context–Aware Systems (CAS) base their fall detection procedure on the signals provided by fixed sensors (cameras, microphones, vibration sensors, etc.) that are located in the physical environment around the user to be monitored (normally

CAS architectures, which encompass both vision-based and ambient-based solutions, pose several technical and economical problems. Firstly, the monitoring area where the patient (or user) can be tracked is confined to the space (e.g. a room) where the environmental sensors are deployed. Within this constrained ‘tracking zone’, the quality of the fall detection may be deteriorated by uncontrollable events (such as the alteration of the illumination levels, spurious sounds, shadow zones originated by the unexpected displacement of the furniture, tumbling objects, etc.). In addition, the operations required by the development of a CAS-based solution (including the installation, adjustment and maintenance of the sensors) normally result in a non-negligible cost. Furthermore, the use of video cameras to supervise the movements may affect the sense of privacy of the patients.

On the other hand, wearable FDS follow a different approach to monitor and identify fall patterns in the user activity. Wearable systems utilize accelerometers (and other mobility sensors) which are inserted in the clothes or carried by the patients as personal garments or gadgets. Wearable FDS directly sense the physical variables that characterize the user’s movements with independence of the user’s position or surrounding environment. In fact, wearable FDS are usually provided with wireless communication interfaces (e.g. 3G/4G mobile cellular connection to the Internet), which enable the remote monitoring of the patient status almost ubiquitously. In this regard, smartphones are a good candidate to implement wearable FDS as long as they natively integrate a wide diversity of mobility sensors (accelerometer, gyroscope, magnetometer) and support multi-interface wireless communications (Wi-Fi, 3g/4G, Bluetooth). The increasing computing power, battery and memory capacities of current smartphones allow deploying complex algorithms aimed at detecting falls in real time. Hence, a personal device (which is omnipresent in the daily life of many citizens) can be transformed in a FDS by simply loading a software code (an *app*) in a smartphone and without requiring any specific hardware. This rapid and cost-effective way of developing a FDS on a massively popular personal device has fostered the apparition of an extensive research literature dealing with smartphone-based FDSs during the last five years. Most studies propose solutions where the smartphone is the sole element of the architecture. In these ‘stand-alone’ systems, all the functionalities (sensing, communication, algorithm computation, alarming, etc.) reside in the smartphone, which generates the system decision based on its own embedded sensors and without any support from any other external

This work was supported by European FEDER funds and the Spanish Ministry of Economy and Competitiveness [grant TEC2013-42711-R] (URL: http://www.mineco.gob.es/portal/site/mineco/)

Authors are with the Department of Electronics Technology (Departamento de Tecnología Electrónica), Universidad de Málaga, Andalucia Tech, Malaga, Spain (corresponding author: Eduardo Casilari, phone: 24-952132755; fax: 34-952131447; e-mail: ecasilari@uma,es).

component.

Different studies [6] [7] have shown that placing the wearable FDS near the chest and the waist notably increases the accuracy of the fall prediction as long as these locations are closer to the to the center of mass of the human body than other conventional positions for a smartphone. In fact, the typical use of a pocket (e.g. in a shirt or in a trouser) normally provokes the smartphone to swing freely, which leads to a worse characterization of the human movements with the embedded accelerometers of the smartphone. Consequently, in order to boost the effectiveness of the fall detection decision, some works in the literature propose to attach the smartphone to the chest or waist by means of a flexible band or a similar fixing belt or strip. However, the fastening of the phone undoubtedly reduces the ergonomics of the FDS whereas it hinders the conventional use of the smartphone (making calls, messaging, web surfing, etc.).

In order to cope with these problems of patient discomfort that the placement of a smartphone-based FDS can produce, smartwatches have been proposed as an alternative solution to deploy wearable FDSs [8]–[10]. Nowadays, commercial smartphones are programmable devices which allow installing user-defined mobile applications and which may also integrate mobility sensors and wireless standardized interfaces. Smartwatches clearly improve the physical ergonomics of the FDS as they permit a more natural attachment of the accelerometer to the patient’s body. Contrariwise, the main drawback of a FDS merely based on the information captured by a smartwatch is that the movements of the wrist do not always reflect the global stability of the body. Thus, a brusque or impulsive gesture performed with the hands or the arms may be misidentified by the detection algorithm as a fall (originating a ‘false positive’).

Google’s Android is by far the most widespread smartphone Operating System (OS), leading with a 82.8% market share in May 2015 [11]. Consequently, to date almost all the research about smartphone and smartwatch-based FDS has selected Android as the mobile OS to develop prototypes and experimental testbeds. [12].

Aiming at achieving a higher confidence in the identification of falls, we describe a FDS that combines both a smartwatch and a smartphone. The simultaneous use of the signals captured by the built-in sensors of the smartphone and an external (normally Bluetooth-enabled) accelerometer has been considered in papers such as [13], [14], [8]. The interesting system presented in [15] also proposes to employ a fall detection algorithm in a smartwatch to confirm the fall detection decision made by an app in a Smartphone. However, to the best of our knowledge, the particular performance improvements in the detection accuracy accomplished by the combined use of the two sensors have not been systematically assessed before by other authors.

II. Architecture of the System

As it has been mentioned before, the proposed system, (portrayed in Figure 1 and also described in [16]), consists of two basic components: a smartwatch and a smartphone, both provided with Android Operating System and built-in mobility sensors (a triaxial accelerometer and a gyroscope).

The employed smartwatch was a LG W110 G Watch R model, featuring 512 MB of RAM, 1.2GHz Qualcomm Snapdragon 400 MSM8226 1.2 GHz processor, 410 mAh battery capacity and 4 GB of internal storage. As it refers to the smartphone, the smartphone model utilized in the testbed was a LG Nexus 5. This phone includes a Qualcomm Snapdragon 800 2.26 GHz processor, 2 GB of RAM and a 2300 mAh battery. Similar results were obtained with other smartphone models.

Each device runs its own fall detection algorithm separately. To this end, an Android application (app) implementing four different algorithms was developed and loaded in the smartphone and the smartwatch. When the system is operating, the two apps independently evaluate the mobility of the users by analyzing the data which are periodically captured with their own sensors (the embedded accelerometer and gyroscope).

In the literature we can find examples [17][18] of Body Area Networks, intended for healthcare applications, which employ low-power wireless standards (such as 802.15.4/ZigBee, 802.15.6 or Ultra-low power Wi-Fi). However, these standards are not incorporated nor supported by the vast majority of commercial smartwatches or smartphones. Thus, in our system, the internal communications between the smartwatch and the smartphone are deployed via Bluetooth. The poor scalability of conventional Bluetooth networks (with piconets of up to 7 slaves) does not pose any problem for the proposed architecture, as far as the systems comprises just two elements and no further sensing nodes are required. In addition, the Bluetooth stack provides different mechanisms (authentication, confidentiality, authorization, etc.) to secure the communications, which is a key point for the acceptability of any real time wireless health monitoring application.

When the app running in the smartwatch detects a fall occurrence, it transmits a specific message to the app in the smartphone through a Bluetooth connection. However, the system only assumes that the fall has actually taken place if the app in the smartphone also identifies a fall pattern within a short period of 1 s before or after the reception of the alerting message from the smartwatch. If the fall is identified in both devices, the smartphone activates an acoustic alarm. If the patients does not manually switch off this alert before 20 s, the app in the smartphone initiates an automatic emergency call (or sends an SMS with a predefined text message) to a preset contact number.

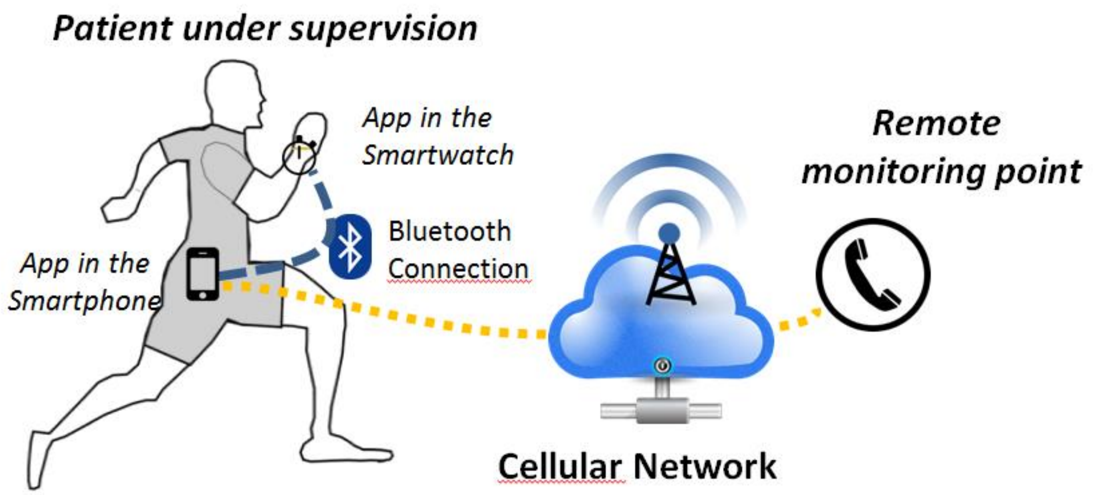

Fig.1. Structure of the hybrid Android System for fall detection.

As it can be inferred, the main concern of the system is to reduce the misidentification of conventional movements as falls. As a consequence, the procedure for remote alerting is not triggered if just one device presumes that a fall has occurred. This avoids typical false positives provoked by situations such as the accidental drop of the phone or a sudden movement of the hands while the body is perfectly stable.

I. Evaluated Fall Detection Methods

The developed Android apps are intended to compare four different fall detection algorithms that make their decision as a function of the signals measured by the built-in accelerometers and gyroscopes of the wearable devices. As the smartwatch presents constrained computational and storage resources, we did not consider other complex fall pattern recognition approaches (such as those based on artificial intelligence, rule-based or machine learning techniques) that have been utilized by the research literature (see [19] or [12] for a comprehensive state-of-the-art). Thus, we compared four basic 'thresholding' algorithms, which only assumes a fall if one or several mobility variables surpass some decision thresholds (simultaneously or in consecutive observation intervals). The four implemented algorithms, which have been also compared in [20] (with smartphone-only architecture) are described in the following sub-sections:

A. Basic Threshold Monitoring

Falls typically provoke the presence of unexpected peaks of the body acceleration. So, according to a basic thresholding method, a fall is assumed whenever the module of the acceleration (or *SMV*, Signal Magnitude Vector) exceeds a certain threshold (SMV_{Th}). The value of SMV_i (for the *i-th* measurement of the acceleration module) can be calculated as:

$$SMV_i = \sqrt{\left|A_{x_i}\right|^2 + \left|A_{y_i}\right|^2 + \left|A_{z_i}\right|^2} \quad m/s^2 \qquad (1)$$

where A_{xi}, A_{yi} and A_{zi} refer to the three acceleration components for that *i-th* sample in the direction of the *x*, *y*, and *z*-axis, respectively. These components are periodically measured by the tri-axial accelerometer embedded in the smartphone or the smartwatch.

B. Fall Index

This method, presented by Yoshida in [21], continuously computes a certain Fall Index (*FI*), which is compared with a certain decision threshold (FI_{Th}). This FI_i index (for the *i*-th measurement interval) can be calculated based on the evolution of the last 20 samples of the three components of the acceleration:

$$FI_i = \sqrt{\sum_{k=x,y,z} \sum_{i-19}^{i} \left(A_{k_i} - A_{k_{i-1}}\right)^2} \qquad (2)$$

where the sub-index *k* indicates the direction (*x*,*y*,*z*) od the corresponding measured acceleration component.

This method tries to avoid the false positives generated by the basic thresholding technique when the body carries out some kind of brusque movements. Conversely, the algorithm is less robust in the presence of 'slow' falls, which may remain undetected.

C. Two-phase detection method

This method is a variant of PerfallD algorithm described in [20]. The method performs the detection by monitoring the SVM and the module of acceleration at the absolute vertical direction ($|A_{vi}|$), which can be computed (for the *i*-th sample) as:

$$\left|A_{v_i}\right| = \left|A_{x_i} sin\theta_{z_i} + A_{y_i} sin\theta_{y_i} - A_{z_i} cos\theta_{y_i} cos\theta_{z_i}\right| \qquad (3)$$

In the previous formula θ_{yi} and θ_{zi} denote the measured values of the pitch and roll angles (at the *i-th* sampling interval), sensed by the gyroscope which is integrated in the wearable devices (smartphone and smartwatch).

The method divides the study of the movements into two stages or phases: free fall and impact. In order to distinguish the acute decay of the acceleration module originated by a Free Fall (FF), the algorithm constantly examines if the absolute maximum difference of the captured values of SMV_i within a short time window (*winFF*) surpasses a triggering threshold (SMV_{FF}). If that condition holds, then the system triggers the recognition of the Impact Phase (IP). During this second stage, the algorithm calculates the difference between the maximum and minimum values of SVM_i within a second observation time window (whose duration is set to win_{IP}). If this difference rises above another (higher) detection threshold (SMV_{IP}), which could indicate that the patient's body has hit the floor, the system suspects that a fall may have taken place. A similar criterion is applied in parallel with the values of $|A_{vi}|$ and the corresponding thresholds AV_{FF} and AV_{IP}. A fall is only detected if the two phases and the two detection conditions are simultaneously satisfied for *SVM* and $|A_{vi}|$.

D. iFall method

This method [22] also contemplates that a fall initiates with a sudden decrease of the acceleration module. After this brusque free-fall-phase, the impact against the floor produces a sharp peak of the acceleration. Consequently, the method assumes a fall occurrence if the value of SMV_i consecutively goes beyond a lower (SMV_l) and an upper threshold (SMV_u) during a pre-set observation time window (win_O). In any case, the method only alerts about the fall if the user changes his/her position from a vertical to a horizontal posture. So, if the patient does not return to the vertical position within a second "post-fall" (PF) observation time window (with a duration of win_{PF}), the system reports the detection. Otherwise, the possible fall is neglected.

II. RESULTS

The proposed system was systematically evaluated with a series of movements: mimicked falls and ordinary Activities of Daily Living (ADLs). The movements were executed by 4 different experimental subjects (healthy males, aged between 22 and 29 years and 165–180 cm tall with an average weight of 67.5kg) in an indoor scenario (a domestic living room). In particular, the volunteers emulated three categories of falls (forward, lateral and backward falls). Besides, ADLs comprised three types of repetitive actions: walking, standing from sitting (and vice versa) and others (including conventional movements such as making gestures with the arms, running, turning the body, or answering the phone). The experiments were repeated ten times per text subject for every type of fall and ADL and for every evaluated detection algorithm. In all the experiments the smartphone was transported within a trouser (in a pocket next to the thigh of the right leg) whereas the subjects wore the smartwatch on their right wrist. This configuration prevents the smartphone from being attached to the chest or the waist. This close connection of the smartphone to the user body (which could not be admissible for a real patient) is employed by many testbeds presented by the literature in order to obtain a better performance of the FDS.

Aiming at evaluating the advantages of a combined scheme, all the tests were iterated to contrast the performance of the architecture with the cases in which just one single device is utilized to monitor the user mobility and to produce the detection decision.

As in most papers of the related literature, the goodness of the system (with one or two devices) to differentiate falls from normal ADLs was evaluated by computing the ratio between the number of true positives (falls that are correctly categorized) and false negatives (i.e. actual falls that were not detected by the system) as well as the ratio between the number of true negatives (ADLs that do not generate any fall alarm) and false positives (ADLs that were mistakenly recognized as falls).

In particular, after observing the response of the system to the executed movements, we calculated the values of the *sensitivity* and *specificity*, two metrics commonly considered to assess the performance of pattern recognition systems with binary classification.

Sensitivity and *specificity* evaluate the ability of the system to properly identify falls and ADLs, respectively. Mathematically, they can be defined as the following success rates:

$$Sensitivity = \frac{TP}{TP + FN} \tag{1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

where *FN* and *FP* represents the amount of false positives and false negatives, whereas *TP* (True Positives) and *TN* (True negatives) designate the number of actual falls and ADLs that have been properly identified, respectively.

An initial 'tuning' test is conducted before the final evaluation to parameterize the four algorithms. In particular, the decision thresholds were selected to achieve an adequate trade-off between the occurrence of false negatives and false positives. The values for the thresholds and observation intervals of the four detection algorithms were set as it follows: SMV_{Th}=25 m/s^2 (for basic thresholding), FI_{Th}=46 m/s^2 (for *Fall Index* algorithm), win_{FF}=0.1 s, win_{IP}=1 s, SMV_{FF}=7.5 m/s^2, SMV_{IP}=18.5 m/s^2 AV_{FF}=6.5 m/s^2 and AV_{IP}=16.5 m/s^2 (for the two-phase detection), win_O=1 s, win_{PF}=20 s, SMV_l=2.5 m/s^2 and SMV_u=24 m/s^2 (for *iFall* algorithm)

Table 1 (showing the measured sensitivity) and Table 2 (indicating the specificity) summarize the results of the tests for the four considered algorithms and the different typologies of emulated falls and ADLs. The tables allow comparing the performance of the FDS when the hybrid detection is operated (i.e. when the fall alarm requires both the smartphone and the smartwatch to identify a fall pattern simultaneously) and the performance of the cases when just one device (the smartphone or the smartwatch) is utilized to detect the falls.

Table 1 2 illustrate that the use of a hybrid system (i.e. that combining a smartphone and a smartwatch) outperforms the specificity of the systems with just one active device in the range of 3-15% for the four considered algorithms. This can be justified by the fact that false positives caused by one device are compensated by an adequate detection of the other wearable device. Excluding the case of employing the basic thresholding algorithm, which is too simple to achieve an effective detection in both devices simultaneously, this specificity gain is accomplished just at the expense of a small loss in the sensitivity metric (see Table 1).

The last column in Table 1 indicates the difference between the best and the worst case of the measured sensitivity taking into account the different types of falls that have been emulated. In Table 2 that column includes the same maximum difference for the case of the measured sensitivity for the different typologies of ADLs that were executed in the experiments. In this sense, these results show that the proposed hybrid architecture presents a more homogeneous behavior than the architectures where just a single device is utilized to detect the falls. In these smartphone-only (or smartwatch-only) based schemes, there are typologies of movements (e.g. those classified as 'other' in Table 2) for which a poor specificity of 60% is attained. This variability is clearly reduced by the combined scheme (which reaches a minimum specificity of 80% for the worst typology).

In any case, the actual importance of the specificity (when evaluated in a testbed with systematic ADL movements) should be revised. In most studies in the literature, the parameterization of the detectors (e.g. the detection thresholds) is aimed at achieving a trade-off between sensitivity. And specificity. However, this compromise between false negatives and false positives should be revisited. A value of 95% for the sensitivity can be considered an admissible metric for a good fall detector (as long as only 1 out of 20 falls will be unnoticed). However, the same value of 95% for the specificity can be completely inacceptable for a real user as far as 1 out of 20 (5%) of ADLs will be identified as a fall (provoking an annoying alarm that the patient will have to deactivate manually or an alert that will misinform the remote monitoring user). So, one of the crucial questions about a fall detector is: how many false alarms may the system cause daily? We analyzed this practical metric with our hybrid system by computing the number of false positives detected after monitoring one of the volunteers, who wore the system with the two devices during 24 hours of everyday life. Table 3 shows that for the four algorithms (even with the combined architecture) several false positives were registered.

Another key factor of a system supported by apps on Android-based devices is battery consumption. The constant reading of the built-in sensors, the Bluetooth transmissions and the computation in real-time of the detection algorithm can make the system unviable if it achieved at the cost of a rapid power depletion in the wearable units.

In order to assess the consumption of our proposed FDS, we performed a set of periodical measurements of the status of the initially fully-charged batteries when the fall detection apps are running in the wearable devices. The evolution of the battery discharge in the smartphone and the smartwatch as a function of the operation time of the system are depicted in Fig.2. To isolate the effect of the FDS in the power drain, no other application is executed in the devices during the measurements. As it can be observed in the figure, the tests were repeated for the four detection algorithms, although no significant differences were found.

The figure 2 illustrates that the influence of the FDS on the battery of the smartphone is almost negligible as the battery level is over 95% of its initial value after 7 hours of continuous operation. On the contrary, the power drain in the smartwatch is much more intense. In fact, more than 50% of the battery is discharged during the same interval of 7 hours. The reduced autonomy of the smartwatch is an aspect that could affect the feasibility of the FDS. Wearable FDS should guarantee a battery duration of at least 16-24 hours so that they can be recharged during the patient's sleep.

III. CONCLUSIONS

Android-based personal devices with embedded mobility sensors allow the software development of automatic Fall Detection Systems at no cost.

This paper has presented and evaluated a prototype of an architecture for fall detection that incorporates two personal popular devices: a smartphone and a smartwatch, which intercommunicate with Bluetooth, which is natively supported by most current commercial smartwatches and smartphones.

. In order to reduce the occurrence of false positives, the proposed system only assumes that a fall has occurred when it is detected at the same time by both devices.

The system was tested against systematic experiments consisting of emulated falls and ADLs executed by volunteers. The obtained results indicate that, even with simple threshold-based fall detection algorithms, the j use of the two devices notably increases the specificity of the system (the capability to discriminate ADLs correctly) just at the expense of a small decay in the sensitivity (the efficiency to identify falls). The real specificity of the system is evaluated by monitoring an experimental subject during a period of 24 hours, showing that an apparently acceptable value of the specificity (obtained with mimicked ADLs) can lead to a not-negligible number of false alarms per day when the system is employed in a realistic environment.

The present hybrid FDS does not introduce any specific or bulky wearable component. Moreover, in contrast with other smartphone-based FDS the user is not obliged to carry the smartphone in an unnatural position to increase the efficiency of the detection. In the proposed architecture, thanks to fact that the detection decision is based on two components, the smartphone can be transported in a more comfortable way

The study is completed by investigating the consumption of the batteries in the devices. The performed analysis shows that the battery restrictions in the smartwatch may still affect the

In addition, as any other system meant for elderly patients, the proposed FDS requires further and thorough studies on ergonomics and usability.

REFERENCES

[1] World Health Organization. Ageing & Life Course Unit, "WHO global report on falls prevention in older age," World Health Organization, 2008.

[2] S. Yoshida, "A Global Report on Falls Prevention Epidemiology of Falls," World Health Organization, 2007.

[3] CDC (Centers for Disease Control and Prevention), "Costs of falls among older adults," *Report of the Injury Center*, 2014. [Online]. Available: http://www.cdc.gov/homeandrecreationalsafety/falls/fallcost.html. [Accessed: 15-Feb-2016].

[4] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection-principles and methods," in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, 2007, pp. 1663–1666.

[5] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[6] S.-H. Fang, Y.-C. Liang, and K.-M. Chiu, "Developing a mobile phone-based fall detection system on android platform," in *Proceedings of the Computing, Communications and Applications Conference (ComComAp)*, 2012, pp. 143–146.

[7] L. Kau and C. Chen, "A Smart Phone-based Pocket Fall Accident Detection, Positioning and Rescue System," in *IEEE International Symposium on Bioelectronics and Bioinformatics (ISBB)*, 2014, pp. 1–4.

[8] I. Maglogiannis, C. Ioannou, G. Spyroglou, and P. Tsanakas, "Fall Detection Using Commodity Smart Watch and Smart Phone," in *Artificial Intelligence Applications and Innovations*, I. L. Maglogiannis, H. Papadopoulos, S. Sioutas, and C. Makris, Eds. Springer, 2014, pp. 70–78.

[9] H. Honglun, H. Meimei, and W. Minghui, "Sensor-based wireless wearable systems for healthcare and falls monitoring," *Int. J. Smart Sens. Intell. Syst.*, vol. 6, no. 5, pp. 2200–2216, 2013.

[10] T.-H. Tan, M. Gochoo, C.-S. Chang, C.-T. Wu, and J. Y. Chiang, "Fall Detection for Elderly Persons Using Android-Based Platform," in *Proceedings of the 8th International Forum on Strategic Technology (IFOST 2013)*, 2013.

[11] IDC, "IDC Worldwide Mobile Phone Tracker (May 2015)." [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed: 19-Feb-2016].

[12] E. Casilari, R. Luque, and M. Morón, "Analysis of android device-based solutions for fall detection," *Sensors*, vol. 15, no. 8, pp. 17827–17894, 2015.

[13] W.-J. Yi and J. Saniie, "Design flow of a wearable system for body posture assessment and fall detection with android smartphone," in *Proceedings of the IEEE International Technology Management Conference (ITMC)*, 2014, pp. 1–4.

[14] A. Boehner, "A Smartphone Application for a Portable Fall Detection System," in *Proceedings of the National Congress of Undergraduate Research (NCUR)*, 2013.

[15] T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lones, J. N. Slettevold, and S. M. Weggersen, "A Combined Smartphone and Smartwatch Fall Detection System," in *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, 2015, pp. 1443–1448.

[16] E. Casilari and M. A. Oviedo-Jiménez, "Automatic fall detection system based on the combined use of a smartphone and a smartwatch," *PLoS One*, vol. 10, no. 11, p. e0140929, 2015.

[17] F. Martelli, C. Buratti, and R. Verdone, "On the performance of an IEEE 802.15. 6 wireless body area network," in *Proceedings of the 11th European Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless)*, 2011, pp. 1–6.

[18] T. Hayajneh, G. Almashaqbeh, S. Ullah, and A. V Vasilakos, "A survey of wireless technologies coexistence in WBAN: analysis and open research issues," *Wirel. Networks*, vol. 20, no. 8, pp. 2165–2199, 2014.

[19] M. A. Habib, M. S. Mohktar, S. B. Kamaruzzaman, K. S. Lim, T. M. Pin, and F. Ibrahim, "Smartphone-based solutions for fall detection and prevention: challenges and open issues," *Sensors*, vol. 14, no. 4, pp. 7181–7208, 2014.

[20] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," in *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 292–297.

[21] T. Yoshida, F. Mizuno, T. Hayasaka, K. Tsubota, S. Wada, and T. Yamaguchi, "A wearable computer system for a detection and prevention of elderly users from falling," in *Proceedings of the 12th International Conference on Biomedical Engineering (ICBME)*, 2005.

[22] F. Sposaro and G. Tyson, "iFall: an Android application for fall monitoring and response," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009)*, 2009, pp. 6119–6122.

Table 1. Obtained **sensitivity** for the different types of emulated falls, employed detection algorithm and considered devices

Algorithm	Employed Device	Type of Fall			Global	Maximum deviation between fall types
		Forwards	Backwards	Lateral		
Basic Threshold	**Smartphone & Smartphone**	**0.85**	**0.70**	**0.80**	**0.78**	**0.07**
	Only Smartphone	1.00	1.00	0.90	0.97	0.10
	Only Smartwatch	1.00	0.90	0.80	0.90	0.20
Fall Index	**Smartphone & Smartphone**	**0.95**	**0.90**	**0.90**	**0.92**	**0.05**
	Only Smartphone	0.80	1.00	1.00	0.93	0.20
	Only Smartwatch	0.90	1.00	1.00	0.97	0.10
Two-phase	**Smartphone & Smartphone**	**0.95**	**0.90**	**0.90**	**0.92**	**0.05**
	Only Smartphone	1.00	0.90	0.90	0.93	0.10
	Only Smartwatch	1.00	0.90	1.00	0.97	0.10
iFall	**Smartphone & Smartphone**	**1.00**	**0.95**	**0.95**	**0.97**	**0.05**
	Only Smartphone	0.90	0.90	1.00	0.93	0.10
	Only Smartwatch	1.00	1.00	0.90	0.97	0.10

Table 2. Obtained **specificity** for the different types of emulated ADLs, employed detection algorithm and considered devices

Algorithm	Employed Device	Type of Fall			Global	Maximum deviation between fall types
		Walk	Sit/Stand	Other		
Basic Threshold	**Smartphone & Smartphone**	**1.00**	**1.0**	**0.80**	**0.93**	**0.20**
	Only Smartphone	1.00	0.90	0.60	0.83	0.40
	Only Smartwatch	1.00	1.00	0.70	0.90	0.30
Fall Index	**Smartphone & Smartphone**	**1.00**	**0.95**	**0.80**	**0.92**	**0.20**
	Only Smartphone	1.00	1.00	0.60	0.87	0.40
	Only Smartwatch	1.00	0.80	0.60	0.80	0.40
Two-phase	**Smartphone & Smartphone**	**1.00**	**1.00**	**1.00**	**1.00**	**0.00**
	Only Smartphone	0.90	1.00	0.80	0.90	0.20
	Only Smartwatch	1.00	1.00	0.80	0.93	0.20
iFall	**Smartphone & Smartphone**	**1.00**	**1.00**	**0.95**	**0.98**	**0.10**
	Only Smartphone	1.00	1.00	0.80	0.93	0.20
	Only Smartwatch	0.90	1.00	0.80	0.90	0.20

Table 3. Number of false positives detected after 24 hours of continuous monitoring.

Algorithm	No. of false positives
Basic Threshold	5
Fall Index	4
Two-phase	2
iFall	4

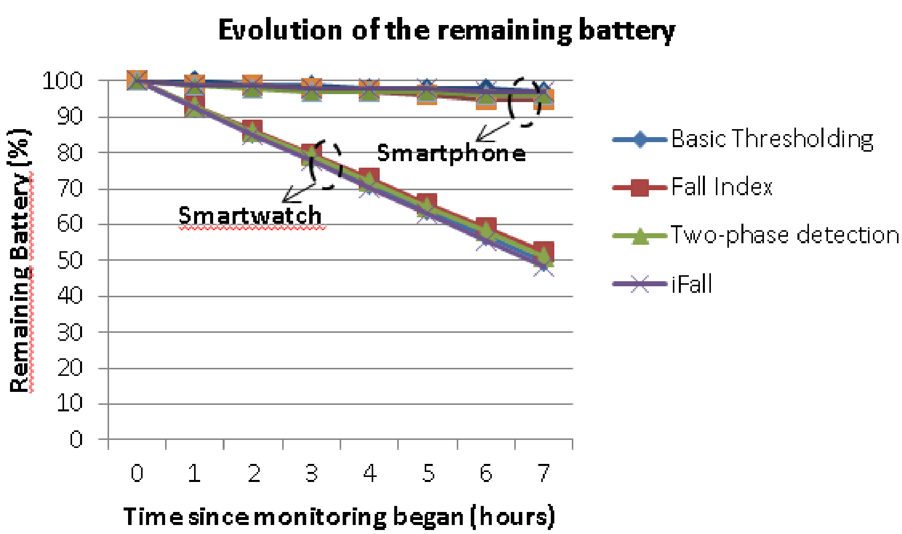

Fig. 2. Evolution of the remaining battery in the smartphone and the smartwatch as a function of the FDS operation time.