

# Using Cloud-assisted Body Area Networks to Track People Physical Activity in Mobility

Giancarlo Fortino<sup>†</sup>, Raffaele Gravina<sup>†,1</sup>, Wenfeng Li<sup>°</sup>, Congcong Ma<sup>°</sup>

<sup>†</sup> Dept. of Informatics, Modelling, Electronics and Systems, University of Calabria, Rende, Italy

<sup>°</sup> Dept. of Logistics Engineering, Wuhan University of Technology, Wuhan, P.R. China

g.fortino@unical.it, rgravina@deis.unical.it, liwf@whut.edu.cn, macc@whut.edu.cn

**Abstract**—This paper describes a novel BSN-based integrated system for detecting, monitoring, and securely recording human physical activities using wearable sensors, a personal mobile device, and a Cloud-computing infrastructure supported by the BodyCloud platform. An integration with a smart-wheelchair system is also presented. BSNs are a key enabling technology for the revolution of personal-health services and their integration with Cloud infrastructure can effectively supports the diffusion of such services in our daily life. Many of these personal-health systems - regardless of their final aim - are based, use or are supported by contextual information on user's physical activity (body posture, movement or action) being performed. This work, hence, aims at providing a basic physical activity service that is capable of supporting personal, mobile-Health applications with real-time activity recognition and labeling both on the personal mobile device and on the Cloud.

**Keywords**—activity recognition, cloud, BSN, wearable sensors

## I. INTRODUCTION

Body Sensor Networks (BSNs) [1] are Wireless Sensor Networks (WSNs) specifically tailored for monitoring human physiological signals. BSNs and their related smart wearables are demonstrating to play a disruptive role in several human-centric applications, such as mobile-Health, e-Fitness and e-Sport, mass events/trends detection and management. BSNs involves smart wearable sensor devices - heterogeneous in terms of form factors, sensing, computation and communication capabilities - that are designed to acquire physiological signals such as electrocardiogram (ECG), heart and respiratory rate, limb and full-body movements, body temperature, blood oxygenation, glucose level, and electromyography (EMG). In addition, such signals are processed to obtain indirect measurements and infer high level information that are used to build diversified non-invasive, 24/7 application services, including activity monitoring, early detection of heart conditions, emotion detection, neuro-degenerative disease monitoring.

Although these technologies are quickly taking place in the market and in our daily life, their actual potential is still partially uncovered. Particularly referring to community-awareness and crowd-sourcing opportunities, there is the need to deal with management of a large number of cooperative and non-cooperative BSNs, which is a hot research topic today. Indeed, enabling pervasive systems based on large communities of users represents a critical success factor for these technologies. The massive amount of data and information that BSN networks can generate at different layers, require efficient, scalable, distributed and interoperable platform for their collection, secure storage, and management. Such tasks simply cannot be achieved relying on the limited resources provided by the BSN alone.

With the aim of providing a solution to this complex task, we have realized, and released in open-source, the SPINE (Signal Processing in Node Environment) framework [2], [3], [4], [5] (a software platform to support BSN sensing and processing management) and BodyCloud [6], [7], [8], a Cloud computing infrastructure tightly integrated with BSN-based platforms, which enables several important functionalities in this context, such as:

- heterogeneous wearable sensing through personal mobile devices acting as local coordinators;
- scalability in terms of processing power for diverse medical analysis;
- scalability in terms of physiological signals collection and data storage;
- global access to processing and storage functionalities;
- simple and authenticated sharing of results.

SPINE is a software framework for the design and fast prototyping of Wireless Body Sensor Network (BSN) applications. It enables efficient implementations of signal processing algorithms for analysis and classification of sensor data through libraries of processing functionalities, and includes an application-level communication protocol too.

BodyCloud is a Software-as-a-Service (SaaS) platform for real-time storage, online and offline management of human vital signs and other physiological sensory signals. Additionally, BodyCloud includes mining functionalities to enable high-level decision-support from sensory data generated by the users' BSNs.. Although our framework is independent from MAC or network stacks, adopting specific solutions can help obtain high performance and reliable communications [9].

We adopted our technologies to realize several e-Health services [10], [11], [12], so showing their effectiveness and flexibility.

In this paper we describe another application of the BodyCloud middleware: an advanced human activity monitoring system, namely *Activity-aaS*, extending our previous work [13].

*Activity-aaS* is a full-fledged hardware/software system to support real-time, non-invasive human activity recognition and monitoring. It also features an accurate and advanced step-counter component. It uses two wearable motion sensors and a personal mobile device where a graphical application provides instantaneous feedbacks to the user; in addition, when Internet connectivity is available, data are also sent to a cloud back-end that serves as long-term, multi-user data storage and processing. Finally, a web-based interface gives remote access to such information at authenticated and authorized users.

In addition, we integrated *Activity-aaS* with a smart-wheelchair system.

The rest of the paper is structured as follows. Section II discusses related work on BSN-based human activity recognition and monitoring. Section III focuses on the proposed physical activity application service. In Section IV an advanced step-counter algorithm is proposed and described in detail. Section V describes the integration of *Activity-aaS* with a smart-wheelchair system. Finally, in Section VI conclusive remarks are drawn and directions of future work are briefly outlined.

## II. RELATED WORK

Automatic physical activity monitoring has attracted tremendous interest and the topic is being studied under very diversified point of views. The problem is addressed with different approaches in terms of sensor types and recognition strategies. One of the most relevant and cited work is [14]. Here, several supervised learning algorithms are used and evaluated to detect physical activities using accelerometer data from sensors placed on different body locations. In [15], authors compared the activity classification accuracy by varying the number and the location of sensor nodes on the human body. In [16] an activity recognition system based on just a single motion sensor node worn at the waist is proposed. In [17], authors investigate the benefits of dynamic sensor selection to achieve the best trade-off between power consumption and activity recognition accuracy and propose an activity recognition method that is associated with an underlying run-time sensor selection scheme. In the last years, thanks to the significant improvements of commercial smartphones (particularly in terms of sensing opportunities), many researchers [18], [19] are highlighting the convenience of implementing physical activity monitoring uniquely relying on users' mobile phones, so to significantly improve user acceptance and reduce economic costs. Interesting surveys on human activity recognition have been published too [20], [21]. For broader reference, interested readers can consult additional interesting survey works on the e-Health domain [22], [23].

## III. ACTIVITY-AASERVICE

*Activity-aaS* takes advantage of the related work targeting a better trade-off among classification accuracy, wearability, power requirements, and programming complexity. It is able to recognize both postures (lying down, sitting, and standing still) and a few movements (walking, running and jumping); furthermore, it includes a convenient guided procedure to extend its recognition capabilities with user-defined activities. Finally, it embeds a simple yet effective accidental fall detection widget that uses the activity classification to determine if a person is unable to stand up after having fallen. Our system uses two wearable accelerometer sensor units based on the Shimmer2R [24] platform (with a pre-installed instance of the SPINE framework) and an Android-based personal mobile device (e.g. a commercial smartphone or tablet) where most of the service logic is actually running. The end-user mobile application is hence written in Android and programmed atop the SPINE-Android framework and assisted by the BodyCloud middleware. Sensor nodes and mobile device communicate over standard Bluetooth.

In particular, BodyCloud has been used for integrating the service with a web application dashboard. Using BodyCloud, all relevant activity-related data are sent to a BodyCloud-based back-end along with time and geo-localization (when available and if granted by the user) references. In addition, with the intent of providing a degree of flexibility in terms computing power demand of the service on the mobile device, we chose to implement three different modes of operation, that can be selected upon application launch.

- 1) *Full Cloud*: the mobile application will only collect the raw data, and send this straight to the cloud. The cloud will then do all needed processing (i.e. feature extraction and classification).
- 2) *Mix Cloud*: the mobile application will be responsible for raw data collection and feature extraction. These features will then be sent to the cloud for classification.
- 3) *Full Local*: all processing will be done on the mobile device. Specifically, raw data collection, feature extraction, and feature classification. The cloud is therefore used only for long-term storage and graphical visualization of statistics (see Fig. 1).

In particular, to reduce network usage, the last operating mode is programmed to communicate with the cloud only after transitions between an activity/posture and another.

According to the BodyCloud programming paradigm based on XML scripting, we defined the following entities for the *Activity-aaS*:

- The *ActivityMonitoring* group, which represents the group of monitored users.
- The *RawAccelerationDataFeed*, *FeatureDataFeed*, and *ActivityDataFeed* modalities. These modalities reflect the three different operating modes described above.
- The *ActivityMonitoring* workflow, which models a three sequential nodes workflow able to read body motion data collected by the reader node, extract the features out of it, and apply the activity classification

TABLE I. POSTURE/MOVEMENT RECOGNITION ACCURACY.

Sitting	Standing	Lying	Walking	Falling
96%	92%	98%	94%	100%

algorithm. Note that this workflow is specifically activated when Activity-aaSvc runs in “Full Cloud” mode.

- The Activity View, which is used to model the web-based graphical representation of the various activities being performed by the user (currently a simple pie chart for statistics visualization).

The activity recognition relies on a classifier that takes accelerometer data features (max, min and average; see above) and recognizes the movements defined in a training phase. A K-Nearest Neighbor (KNN) classifier [25] has been chosen for recognition. The significant feature set has been identified using an off-line sequential forward floating selection algorithm [26]. Experimental results have shown that, given a certain training set, the classification accuracy is not much affected by the K value or the type of distance metric. This is because our target activity classes present high inter-class correlation as well as high inter-class separation. Therefore, K set to 1 and the Manhattan distance are used as parameters of the KNN-based classifier.

The overall recognition accuracy (see Table I) reached by the recognition system is considerably high, reaching an average value of 97%. The ground truth used to measure the system accuracy has been obtained by manual, online annotation of the actual activities being performed. In synthesis, during system evaluation, we asked subjects to wear our system and to freely perform any of the four target activities in a laboratory setting (a 150m<sup>2</sup> room furnished with desks, sofas, chairs). We enabled a dedicated panel on the mobile application with a radio button to let us selecting the actual activity being performed by the subject and we programmed the application to store activity classification outputs on file a simple CSV file with two label columns, one for the actual activity (as indicated using the radio button) and the other for the activity classified by the system. Finally, we analyzed offline this file by counting the number of recorded classifications for each activity and comparing each estimated activity class against the corresponding ground truth value, so to detect classification errors. Therefore, the accuracy percentage of a given activity is obtained as the number of correct classifications divided by the total number of samples recorded on the file for that activity.

Fig. 1 depicts a statistics web-based dashboard of the activities performed by the user. The time period for statistics generation can be freely selected by the user. Specifically, the dashboard currently consists of a pie chart indicating the time percentage spent for each performed activity. If a certain activity is not performed at all within the selected time interval, it will not be show in the chart. In addition, a table reports the same information as the pie chart plus the total time spent (expressed in days.hours.minutes.seconds) within the selected time interval.

#### IV. STEP-COUNTER ALGORITHM

This section describes an innovative step-counter algorithm, whose main design requirements are the followings:

- Use of accelerometer data from a body worn sensor;
- Energy and computation efficient design to support embedded implementations;
- Use of a single sensor node, placed on the waist;
- General-purpose algorithm, to fit walking patterns of healthy people as well as elderly and/or people with disabilities;
- No need for user-specific calibration;
- High average accuracy (*precision*).

This algorithm has been integrated in Activity-aaSvc to provide additional information to the user and caregivers, allowing for accurate step detection and distance covered.

Several real walk data on different subjects have been collected and studied before starting the algorithm design. Subjects were asked to walk naturally, and to increase/decrease the walking speed occasionally. In particular, a single three-axis accelerometer sensor node was placed on the waist while recording. The sensor has been sampled at 40Hz.

Preliminary conclusions from this analysis include that:

- Off-line downsampling to 20Hz showed that the signal is still well characterized;
- Lateral acceleration presents a poor SNR (Signal-to-Noise Ratio) as the waist swing during walking is heavily influenced by noise;
- Frontal (horizontal) and vertical acceleration present very useful signals; they looks roughly sinusoidal, because the waist is interested by vertical accelerations/decelerations when the feet hit the ground, and by horizontal accelerations/decelerations when the body swings frontally while walking;
- Horizontal acceleration signal is generally less affected by noise than the vertical acceleration signal.

A number of step detection approaches were considered and evaluated, eventually converging to the proposed solution. To simplify the development, debugging, and evaluation, the implementations have been initially programmed in *Matlab*. Only *integer-math* computations were used, so allowing for a more straightforward embedded implementation (as the target embedded platform is based on a micro-controller with no hardware support for floating point operations).

A very simple “*fixed threshold-range*” technique has been studied at the beginning. The thresholds were empirically determined. In this approach, a step is detected if the instantaneous raw acceleration stays within the defined range. To remove multiple detections of the same step, the algorithm sleeps for the shortest “step-time”, measured from the available observations. Although the algorithm is very simple, it showed high precision (> 90% in lab experiments) while applied to healthy people, regardless the gender, weight, height, and



Fig. 1. The main dashboard of the Activity-aaSvc web-app module.

walking speed. However, when applied to elderly people, results were significantly poorer, with almost no steps detected. Intuitively, that is due to less pronounced movements of walking elderly people that lead to lower accelerations of the waist.

An enhanced algorithm followed a different approach. Rather than defining *a priori* the threshold range, it is possible to determine the best threshold for the monitored subject by running a search algorithm that requires a set of walking data of that subject, the number of steps inside that recording, and the required accuracy. The threshold search is performed iterating the step-counter algorithm previously described on the walking data, with a decreased threshold (initially set to the upper accelerometer scale value) at each iteration and comparing the number of detected steps against the actual number. The search stops when the required accuracy has been reached or the number of detected steps exceeds the actual number, and the threshold found is returned. This implementation performs overall better than the first one, but requires a training phase and the manual counting of the steps during the training. If the counting is incorrect and/or during the recording a significant number of data packets get lost, the threshold search can be compromised.

The first algorithm does not recognize effectively the steps of elderly people. On the other hand, the second algorithm needs a preliminary training phase. Furthermore, they both work on the raw accelerometer data.

To overcome these limitations, a third algorithm has been defined, borrowing some ideas from the previous ones. At the same time, additional real walk data have been recorded from hospitalized subjects (elderly and people with motor disabilities). This latter algorithm attempts at detecting steps by identifying the decreasing segment (*falling edge*) which

corresponds to the last fraction of a step movement.

A step is characterized by time constraints (i.e. it cannot be “too” fast or “too” slow). However, walk patterns vary from people to people and even for the same person it might change from time to time; hence, the amplitude and slope of the acquired signal can vary significantly.

To simplify the pattern recognition, the raw acceleration is first processed with a *smoothing filter* which removes the high frequency components. Then, it looks for local maximums. When a local max is found, it looks for a local minimum. After the local min is found, the candidate segment (i.e. a potential step) is also identified. Two features are then extracted and used to determine whether the candidate belongs to a real step or not. The candidate is classified as *step* if (i) it has an acceleration drop within a certain range (specified by a “tolerance” parameter around a threshold), and (ii) its time duration is within a certain interval. More specifically, the pre-processing is a 9-point windowed smoothing filter using Gaussian kernels. Because they are applied to a digital signal, the sum of the kernels must be 1. However, because the algorithm works on integer-math, they have been scaled so that decimal factors are removed.

The following Gaussian kernels have been selected:

- { 5, 30, 104, 220, 282, 220, 104, 30, 5 }

The threshold is coarsely initialized, but it is automatically adapted while steps are recognized. In particular, it is continuously updated with the average of the last 10 acceleration drops that were classified as steps.

Since the threshold is just coarsely initialized at the beginning and because even following steps can occasionally present significant variance in their vigor, the updated threshold is

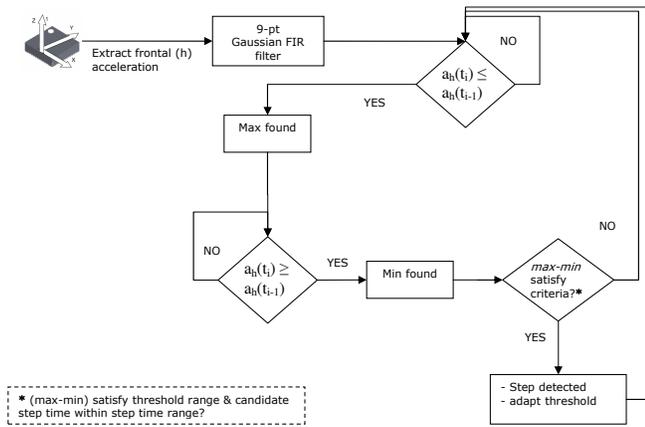


Fig. 2. Block diagram of the step-counter algorithm.

adjusted with parametric coefficients to specify the amplitude range in which a candidate must fall.

Finally, to reduce “false positive” recognitions, e.g. due to sudden shocks or slow tilts of the sensor, the time elapsed between the local max and min (which it is simply determined as the product between the number of samples composing the segment and the sampling period) must be longer than the “*minimum step time*” and shorter than the “*maximum step time*”. Both values have been determined empirically from the available observations. In particular, based on the available walking data, the following values have been identified:

- *minimum step time* = 350ms
- *maximum step time* = 2000ms.

Fig. 2 shows the block diagram of the proposed algorithm.

The proposed algorithm has been initially evaluated on the computer using Matlab, and eventually implemented on a wireless sensor node running SPINE. For this application, the node-side of SPINE has been extended with the proposed algorithm. Every time the node detects a step, it communicates to its coordinator the total number of steps taken so far (this to avoid counting errors in case of lost packets). On the coordinator side, only minor additions were required.

As a running example of the algorithm, Fig. 3 shows the raw data of the frontal waist acceleration during normal walking of a healthy subject, sampled at 20Hz. The “end” of each step (which then brings to the beginning of the following one with the other foot), corresponds roughly to the sharp lower spikes in the plot.

Fig. 4 shows instead the result of the Gaussian filtering; the small black dots in Fig. 4 correspond to the points where the step-counter algorithm has detected steps. It is worth noting that although the strength of the steps sometime change significantly, the algorithm still detects properly all the steps.

The proposed step-counter algorithm has been tested on several subjects, from healthy young people to elderly and/or with motor disabilities. An experimentation consisting of 40 walking sessions from 8 healthy subjects (both males and females, with different height and weight) and 6 hospitalized subjects (with post-stroke disabilities, walking using crutch or

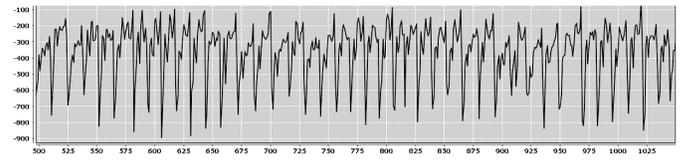


Fig. 3. Raw Data of the frontal (horizontal) acceleration of the waist during normal walking.

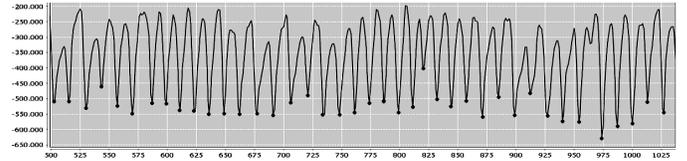


Fig. 4. Result of the Gaussian filtering of the data shown in Figure 3.

wheels walkers, or simply elderly) has been carried out and our step-counter obtained an average precision of 91%.

The worst under-estimation was 27%, while the worst over-estimation was 18%. It is worth noting, however, that the subjects were not video recorded during their walking data collection; hence, it was sometimes difficult to validate the number of reported steps (manually counted by direct observation during walking) in each experiment. Furthermore, we detected limited packet loss that caused discrepancy between the number of reported steps and the recorded ones.

In conclusion, the main contribution of the proposed algorithm is the ability to provide a good estimate of the steps taken by people with disabilities and/or using crutch or wheels walkers. To the best of our knowledge, this is the only system being able to work properly for the latter category of users.

## V. INTEGRATING ACTIVITY MONITORING ON A SMART WHEELCHAIR

A non-motorized wheelchair has been enhanced with four main functionalities: wheelchair overturn detection, activity recognition, body posture recognition, and location tracking. Activity recognition is carried out with our BSN-based system described in Section III, while we designed a customized hardware platform for supporting the other functionalities.

This platform is composed of a sensor module, a signal processing module and a wireless communication module. The signal processing module is based on Arduino, it features low energy consumption, high sampling rate and high processing capabilities. The wireless communication module uses Bluetooth to support full interoperability with Activity-aaSvc. The sensor module uses accelerometers and pressure sensors to detect wheelchair overturns and user posture on the wheelchair cushion, as described in detail in the next subsections.

### A. Overturn Measurement

To detect wheelchair overturns we equipped the smart-wheelchair system with a three-axis, low-power capacitive micro machined accelerometer (Freescale MMA8451). We sample the accelerometer at  $f_s = 20Hz$ , we compute the instantaneous cross-axial acceleration signal energy as follows:

$$E_{tot} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

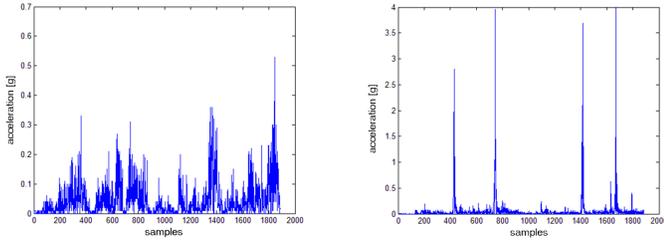


Fig. 5. Wheelchair operating regularly (left) and during four overturn events (right).

and we eventually compute second-by-second averaged  $E_{tot}$  values. Finally, we perform a threshold-based wheelchair overturn detection over the absolute value of the difference between consecutive  $E_{tot}$  values. Specifically, we extract another feature,  $Ev$  (signal energy variation), calculated as follows:

$$Ev_t = |E_{tot,t} - E_{tot,t-1}| \quad (2)$$

The threshold value has been obtained empirically, by monitoring acceleration data of our smart wheelchair moving regularly and occasionally let overturn intentionally (see the plots depicted in Fig. 5).

From Fig. 5, we can see that overturns determine  $Ev$  values significantly higher than 1.5g; it is therefore appropriate to set the  $Ev$  threshold to 1.5g to detect wheelchair overturn events.

### B. Body Posture Measurement

Wheelchair users might feel uncomfortable to retain the same sitting posture for a long time; yet they might have difficulties to change posture. In addition, sitting on the wheelchair for long periods might lead to serious illness such as sores.

We instrumented our wheelchair with a smart cushion [27] that can measure pressure applied by the body weight from multiple points. Specifically, three pressure sensors are placed respectively on the two side and the center of the cushion. We adopted a Force Sensing Resistor (FSR) pressure sensor produced by Interlink Electronics. It is ultra-thin, lightweight, and highly accurate. Its size is approximately  $45 \times 38 \text{ mm}$ . This FSR can sense applied force anywhere in the range of  $100 \text{ g} - 10 \text{ kg}$  ( $g = 9.8 \text{ N/kg}$ ). This pressure sensor is very convenient for our purposes as it can be easily embedded into the cushion textile or foam filling, as shown in Fig. 6. The cushion also includes a 3-color LED indicator and a speaker.

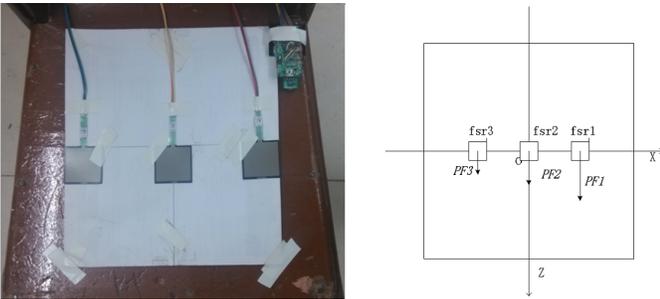


Fig. 6. Smart cushion equipped with FSR pressure sensors (left) and representation of pressure sensors data in coordinate system (right).

TABLE II. WHEELCHAIR USER'S POSTURE RECOGNITION ACCURACY.

No one Sitting	Sit straight	Lean to left	Lean to right
100%	98%	96%	95%

The smart cushion is placed on the chair and it is unobstructive for users. When the user sit on the chair, pressure sensory data are collected by the smart cushion and sent to processing module to recognize whether the user is comfortably sit or leans on a side. The user can set a sitting time alarm and when it is reached, the indicator light will change the color and a sound warning is played.

According to the pressure distribution among the three FSR sensors determined by the body weight, we can analyze the sitting posture, detecting whether the user is sitting upright or swinging from side to side. To perform such analysis, we set the FSR sensors sampling rate at  $f_s = 20 \text{ Hz}$ . The output of each pressure sensor is an integer number in the range of  $[0, 1023]$ . When the user is not sitting, the output value is zero; the pressure value increases as more weight insists on the sensor. We extract the feature vector of pressure sensor  $F_{pr} = [PF1, PF2, PF3]$  and use a Support Vector Machine (SVM)-based classifier to detect four different body postures. Experiments have been carried out in a laboratory setting with 10 subjects having different height and weight and the results of the performance evaluation are reported in Table II.

### C. User location tracking

The smart wheelchair communicates via Bluetooth with the user's personal mobile device where a customized version of Activity-aaSvc is running so to integrate the information coming from the wheelchair and take them in account. An useful functionality that has been implemented directly on the mobile device regards user's geo-localization, as shown with the screenshot in Fig. 7. We know that, if the wheelchair user is in danger (e.g. in case of wheelchair overturn), it is better to know his/her location accurately to give immediate rescue. For this purpose, we use the geographic coordinates provided by the assisted GPS sensor of the mobile device. When the system detects a dangerous situation, user's location is recorded and sent remotely to relatives and caregivers (see Fig. 1).

## VI. CONCLUSIONS

In this paper we presented Activity-aaSvc, a hardware/software system to support real-time, non-invasive human activity recognition and monitoring that is based on the SPINE framework and the BodyCloud middleware. The service can run stand-alone on a personal mobile device, but can also be connected to a cloud back end so to enable remote access to collected data. Another significant contribution of this work is an advanced step-counter algorithm that has been integrated in Activity-aaSvc to allow for accurate step detection on healthy users and most notably on users with several types of motor disabilities, obtaining an average precision of 91%. Finally, we proposed an integration of Activity-aaSvc with a smart-wheelchair system to better assist and monitor wheelchair users. The smart wheelchair is able to detect and report to the cloud overturns, specific user sitting postures,



Fig. 7. Activity-aaSvc customized with user's geo-localization panel.

and geo-localization. Ongoing works are currently devoted to improving our system to monitor not only users individually, but to jointly infer mobility and activity patterns of those they usually interact with (such as family members, friends, and collaborators). We in fact believe that combining BSN data with large-scale awareness can represent a drastic improvement to current (standalone) m-Health systems. Finally, we plan to introduce a different activity recognition method [28] as it appears promising to improve classification accuracy.

#### ACKNOWLEDGMENT

This work has been partially supported by the joint bilateral Italy/China project "Smart Personal Mobility Systems for Human Disabilities in Future Smart Cities" (N. CN13MO7) and the LOCUBIRehab Project (Polo della Salute, Regione Calabria, CUP J24E07000400005). This work has been partially carried out in the framework of the open-source BodyCloud Project.

#### REFERENCES

- [1] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, "Body area networks: a survey," *Mobile networks and applications*, vol. 16, no. 2, pp. 171–193, 2011.
- [2] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, "Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications," *IEEE Trans. on Human-Machine Systems*, vol. 43, no. 1, pp. 115–133, 2013.
- [3] F. Bellifemine, G. Fortino, R. Giannantonio, R. Gravina, A. Guerrieri, and M. Sgroi, "SPINE: A domain-specific framework for rapid prototyping of WBSN applications," *Software Practice and Experience*, vol. 41, no. 3, pp. 237–265, 2011.
- [4] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari, and G. Fortino, "From Modeling to Implementation of Virtual Sensors in Body Sensor Networks," *IEEE Sensors J.*, vol. 12, no. 3, pp. 583–593, 2012.
- [5] SPINE website, "http://spine.deis.unical.it," Jan. 2015.
- [6] G. Fortino, G. D. Fatta, M. Pathan, and A. Vasilakos, "Cloud-assisted body area networks: state-of-the-art and future challenges," *Wireless Networks*, 2014.

- [7] G. Fortino, M. Pathan, and G. D. Fatta, "BodyCloud: Integration of Cloud Computing and Body Sensor Networks," in *Proc. of the IEEE 4th Int'l Conf. on Cloud Computing Technology and Science*, 2012.
- [8] BodyCloud website, "http://bodycloud.dimes.unical.it," Jan. 2015.
- [9] E.Kartsakli, A.Antonopoulos, A.Lalos, S.Tennina, M. D. Renzo, L.Alonso, and C. Verikoukis, "Reliable MAC Design for Ambient Assisted Living: Moving the Coordination to the Cloud," *IEEE Communications Magazine*, vol. 53, no. 1, January 2015.
- [10] G. Fortino, D. Parisi, V. Pirrone, and G. D. Fatta, "BodyCloud: A SaaS approach for community Body Sensor Networks," *Future Generation Computer Systems*, vol. 35, no. 6, pp. 62–79, 2014.
- [11] G. Fortino and R. Gravina, "Rehab-aaSvc: A Cloud-based Motor Rehabilitation Digital Assistant," in *Proc. of the 2nd Workshop ICT for improving Patient Rehabilitation Research Techniques Workshop - REHAB 2014 (PervasiveHealth 2014)*, 2014.
- [12] R. Gravina, A. Andreoli, A. Salmeri, L. Buondonno, N. Raveendranathank, V. Loseuk, R. Giannantonio, E. Seto, and G. Fortino, "Enabling Multiple BSN Applications Using the SPINE Framework," in *Proc. of Int'l Conf. on Body Sensor Networks*, 2010.
- [13] G. Fortino, R. Gravina, and W. Russo, "Activity-aaSvc: Cloud-assisted, BSN-based system for physical activity monitoring," in *9th IEEE Int'l Conf. on Computer Supported Cooperative Work in Design (CSCWD 2015)*, 2015.
- [14] L. Bao and S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Pervasive 2004, LNCS 3001*, pp. 1–17, 2004.
- [15] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. of the Int'l Conf. on Body Sensor Networks*, 2006.
- [16] N. Ravi, D. Nikhil, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proc. of the 17th Conf. on Innovative Applications of Artificial Intelligence*, 2005.
- [17] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors: accuracy-power trade-o by dynamic sensor selection," in *Proc. of the European Conf. on Wireless Sensor Networks*, 2008.
- [18] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine," in *Proc. of the 4th Int'l Conf. on Ambient Assisted Living and Home Care*, 2012.
- [19] M. Shoaib, "Human Activity Recognition Using Heterogeneous Sensors," in *Proc. of the ACM UbiComp Conference*, 2013.
- [20] L. Chen, J. Hoey, C. Nugent, D. Cook, and Z. Yu, "Sensor-Based Activity Recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Review*, vol. 42, no. 6, pp. 790–808, 2012.
- [21] O. Lara and M. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [22] E.Kartsakli, A.Antonopoulos, A.Lalos, S.Tennina, M. D. Renzo, L.Alonso, , and C.Verikoukis, "A Survey on M2M Systems for mHealth: A Wireless Communications Perspective," *Sensors Journal*, vol. 14, no. 10, pp. 18 009–18 052, 2014.
- [23] G. Acampora, D. Cook, P. Rashidi, and A. Vasilakos, "A Survey on Ambient Intelligence in Healthcare," *Sensors Journal*, vol. 101, no. 12, pp. 2470–2494, 2013.
- [24] Shimmer website, "http://www.shimmer-research.com," Jan. 2015.
- [25] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, January 1967.
- [26] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, November 1994.
- [27] C. Ma, W. Li, J. Cao, S. Wang, and L. Wu, "A Fatigue Detect System Based on Activity Recognition," in *Proc. of Int'l Conf. on Internet and Distributed Computing Systems*, 2014.
- [28] D. Mocanu, H. Ammar, D. Lowet, K. Driessens, A. Liotta, G. Weiss, and K. Tuyls, "Factored Four Way Conditional Restricted Boltzmann Machines for Activity Recognition," *Pattern Recognition Letters*, 2015.