Computer Communications xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Using barometric pressure data to recognize vertical displacement activities on smartphones

Q1 Salvatore Vanini*, Francesca Faraci, Alan Ferrari, Silvia Giordano

Information Systems and Networking Institute, Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), via Cantonale, Manno, Switzerland

ARTICLE INFO

Article history:
Received 4 August 2014
Revised 19 February 2016
Accepted 20 February 2016
Available online xxx

Q2 *Keywords:*
Smartphone computing and communication
Activity recognition
Barometer
Sensor data
Inference model
Energy consumption

ABSTRACT

We introduce a novel, efficient methodology for the automatic recognition of major vertical displacements in human activities. It is based exclusively on barometric pressure measured by sensors commonly available on smartphones and tablets. We evaluate various algorithms to distinguish dynamic activities, identifying four different categories: standing/walking on the same floor, climbing stairs, riding an elevator and riding a cable-car. Activities are classified using standard deviation and slope of barometric pressure. We leverage three different inference models to predict the action performed by a user, namely: Bayesian networks, decision trees, and recurrent neural networks. We find that the best results are achieved with a recurrent neural network (reaching an overall error rate of less than 1%). We also show that decision tree classifiers can achieve good accuracy and offer a better trade-off between computational overhead and energy consumption; therefore, they are good candidates for smartphone implementations. As a proof of concept, we integrate the decision tree classifier in an App that infers user activity and measures elevation differences. Test results with various users show an average recognition accuracy rate of about 95%. We further show the power consumption of running barometric pressure measurements and analyse the correlation of pressure with environmental factors. Finally, we compare our approach to other standard methodologies for activity detection based on accelerometer and/or on GPS data. Our results show that our technique achieves similar accuracy while offering superior energy efficiency, independence from the sensor location, and immunity to environmental factors (e.g., weather conditions, air handlers).

1. Introduction

In the last decades, smartphones became the central computer and communication device in people's activities and lives. Current smartphones includes a variety of sensors that can be used for the continuous real-time location-aware monitoring of human activities as well as environmental conditions [1]. This has opened for research that ranged from very sensitive health applications [2] or privacy-concerned proximity solutions [3] up to leisure purposes [4]. Also, exploiting the multiplicity of mobile devices for large collection of measurements is beneficial to multiple fields, from health [5] to smart-cities [6]. Boosted up by smartphone computing and communication, activity recognition is spreading and developing: more and more applications rely on the knowledge of or on the distinction among human activities. Detecting the action a subject is performing can serve many purposes, for example, monitoring a variety of pathological conditions [7], or sending alerts when a potentially dangerous activity is sensed [8], or identifying lifestyle quality [9,10]. Collected information can be valuable for suggesting countermeasures (e.g., stimulating physical activity if a sedentary lifestyle is recognized). Other information, as location, can be inferred (e.g., detecting floor transitions when a subject passes from climbing stairs to standing still [11]). A major challenge in designing an activity recognition system is the user acceptance. If a system invades the private sphere, the user might be reluctant to adopt it. With the rise of the smartphones, a large part of the activity recognition research switched towards wireless sensor measuring with mobile phones [12]. Smartphones have the double advantage of both being equipped with multiple sensors, and being an ubiquitous commercial product. Latest generation of devices are indeed equipped with a rich set of sensors, including accelerometer, barometric pressure sensor, compass, gyroscope, proximity sensor, light sensor, GPS, microphone, and camera. The key capabilities of sensing, computing and communicating are integrated in the universally accepted and always-with-you smartphone [1,13]. For these reasons, the detection of user activities using sensors embedded in a smartphone is gaining a momentum. Traditional methods for tracking activities with

* Corresponding author. Tel.: +41 586666586.
E-mail addresses: salvatore.vanini@supsi.ch (S. Vanini), francesca.faraci@supsi.ch (F. Faraci), alan.ferrari@supsi.ch (A. Ferrari), silvia.giordano@supsi.ch (S. Giordano).

http://dx.doi.org/10.1016/j.comcom.2016.02.011
0140-3664/© 2016 Published by Elsevier B.V.

smartphones mainly rely on integrated accelerometer sensors. However, the difference in the way of performing a well defined activity and of carrying the device, even for the same user, can lead to a very poor accuracy [14]. A drawback of the accelerometer sensor is energy consumption, which (if always on) is significant [15] and is mostly determined by the need of keeping the phone's components active to access sensors results [16]. Emerging methodologies aim at applying multi-sensor data fusion techniques, taking advantage of the abundance of sensors embedded in smartphones and their complementarity [17,18]. Of course, the use of multiple sensors negatively impacts energy consumption. At present, the two major challenges in the accomplishment of good activity detection with smartphones are still energy efficiency and independence from the device's position along human body. In the present work, we define a new class of activities - Vertical Displacement Activities (VDAs) - where major movement is along the vertical axis. Examples of VDAs can be standing, climbing stairs, riding a cable-car, riding an elevator, or jumping. Our methodology shows that it is possible to identify VDAs with very good accuracy relying only on barometric pressure sensors available on off the-shelf smartphones [11]. Barometric pressure sensors have been traditionally used for height estimation by measuring pressure changes [19]. Information derived from accelerometer data and GPS based localization services can be integrated in a second step, only if really needed. Pressure sensing can provide, for example, complementary information to pedestrian dead reckoning. The main advantage is that switching from a sensor to another can extend battery life and optimize the detection analysis. Moreover pressure measurements are totally independent of the phone position.

Our paper is structured as follows. We firstly discuss the literature (Section 2 and the rationale of our work (Section 3). In Section 4 we present our experiments for investigating the characteristics of barometric pressure in different scenarios. We defined four different user dynamics mode ("standing/walking" on the same floor, "climbing stairs", "riding a cable-car", and "riding an elevator"). Then we collected training labelled data on barometric pressure in the corresponding scenarios. We tested three different inference methods to classify trained data. The metric used to choose the best model is a good trade-off between performance and costs. It is widely known that recurrent neural networks are the state of the art in inference models, however their implementation in resource-constrained devices (i.e. smartphones) presents several issues due to their computational needs and their impact on battery lifetime. On the other hand, decision trees and Bayesian networks are less computationally demanding and have less impact on energy consumption. The results presented in Section 5 show that, although the success rate of the Long Short-Term Memory [20] recurrent neural network to classify our barometric pressure data was very high (only 0.9% of errors, on average), the J48 decision tree algorithm also had a very good performance, providing an average recognition rate of about 95%. For all these reasons, J48 algorithm is the best choice for detecting VDAs on smartphones using barometric pressure data. We also directly measured battery consumption when sampling barometric pressure at a constant rate and found that it is negligible. To demonstrate the advantages of using pressure sensors for activity recognition over sensors traditionally used (i.e., accelerometers and GPS), in Section 6 we analyse and compare accuracy, energy efficiency, indoor effectiveness and phone position independence. Finally, in Section 7 – as a use case scenario – we describe an App for Android where both barometer-based approaches to activity recognition and height estimation have been implemented. This App detects user activity using the J48 decision tree algorithm and shows the altitude graph, the current vertical speed and some statistics about the activities performed by the user.

2. Related work

Many studies have focused on the identification of human VDAs, such as standing, walking, climbing stairs and riding up/down an elevator, from sensors data.

Several pieces of work have been performed with the analysis of accelerometer data, as further discussed in Section 6.1. In general, the accuracy of methods based on accelerometers depends on the position of the sensors (or the phone that embeds sensors) and accelerometers are energy-demanding. Kwapisz et al. [21] collect data from a phone's accelerometer for 29 individuals. Data is analysed and two patterns are identified (periodic and non-periodic). Then, they use three classification techniques (decision trees, logistic regression and multilayer neural networks) to predict the user activities. Krishnan and Panchanathan in [22] evaluate the performance of different discriminative classifiers (i.e., Boosted Decision Stumps, Support Vector Machines and Regularized Logistic Regression) to tackle continuous human activity recognition based on accelerometer data. They propose to capture the rate at which the acceleration changes for activities that have a significant amount of motion (like walking, running, etc.), by computing statistical features like mean, variance and correlation on the first order derivative of the acceleration data. The human-activity recognition system proposed in [14] employs a smartphone with a built-in triaxial accelerometer. It uses a combination of statistical signal features, artificial-neural nets and autoregressive modelling to classify activities. The most cited paper about activity detection using accelerometers is [23], where authors (Bao et al.) use wearable accelerometers to classify a variety of every-day activities (including standing, climbing stairs and riding elevator). In [24] barometric pressure data is used in combination with tri-axial acceleration data and tri-axial gyroscope data to train classifiers and recognize child activities. In [25] pressure sensors are used to improve activity recognition based on acceleration data: in this case, authors limit to plot measures of both barometric pressure and acceleration, and to observe that the change in altitude connected to a pressure change can help to provide a more sophisticated algorithm for activity recognition, but they do not propose any algorithm for activity detection. In [26] a dedicated multi-sensor board containing seven different sensors (microphone, visible light phototransistor, 3-axis accelerometer, 2-axis compass, barometer, ambient light, and humidity) is used to collect measurements from twelve individuals, to infer a subject's activity and classify it as sitting, standing, walking, walking up stairs, walking down stairs, riding elevator down, riding elevator up, and brushing teeth. They employ an ensemble of classifiers to select the most useful features and then use those features to recognize the set of human movements. A second layer of Hidden Markov Models (HMMs) combines the outputs of the classifiers to estimate the most likely activity. Results show that three sensors yield the most discriminative information for recognizing activities: the audio, barometric pressure and accelerometer sensors. This information is complementary: audio captures sounds produced during the various activities, accelerometer data is sensitive to the movement of the body, and barometric pressure helps detecting activities connected to height variations, such as riding an elevator or moving up and down stairs. In [27], four sensors (accelerometer, barometer, gyroscope and magnetometer) are employed to accurately recognize a user's mode of motion when a height change is detected. The algorithm developed has shown a good success rate (from 80 to 96%) in discriminating among walking up or down stairs, riding an elevator, and standing or walking an escalator. In very few pieces of work, GPS location data has been used to learn and recognize the activities in which a person is engaged over a period. For example, in [28] the authors extract a person's activities – such as walking, driving a car, or riding a bus – from traces of GPS data,

Please cite this article as: S. Vanini et al., Using barometric pressure data to recognize vertical displacement activities on smartphones, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.02.011

using a probabilistic temporal model that is based on conditional Random fields (CRF) [29]. Similarly, in [30] generic activities typically performed while a user stays at a location, such as work, leisure, sleep, visit, dining, are inferred from GPS data, using Relational Markov Networks. Such approaches suffer from low-level accuracy and flexibility. In [31], Sankaran et al. use only barometer to detect basic user activities such as *idle*, *walking* and *vehicle*. Their algorithm is based on the number and rate of altitude changes derived from the pressure measurements returned by barometric sensors. Our algorithm has a finer-granularity classification of dynamic modes as it can distinguish the transportation mode (i.e., stairs, elevator or cable-car). Furthermore, the algorithm proposed by Sankaran et al. has a low detection accuracy of the walking mode, which forces them to fuse barometer and accelerometer to complement the barometer-based sensing. Using multiple sensors can help improving the detection of specific activities, especially when information from one sensor is insufficient to recognize them. However, a higher number of sensors involves high energy consumption and may increase the computational overhead. In [32] it is shown how pressure sensors data can be effectively used for floor localisation. The authors present an efficient indoor-stay recognition method requiring measurements such as vertical height between floors, current temperature, and atmospheric pressure value at a reference location, to accurately estimate the floor level. It is the only work where energy efficiency is considered, although it is not directly measured. There are also surveys about integrating sensors on garments for activity recognition tasks, but activities detected are more related to body postures and body-parts movements. For example, authors in [33] present a prototype using strain sensors to distinguish upper body postures. Authors in [34] use conductive textile based electrodes that can be easily integrated in garments to detect specific body activities (e.g., shaking head, looking down, speaking, looking left/right, etc.). Textiles have the same advantage of our approach: they have freedom in positioning of sensors. In addition, they provide space availability and are comfortable to wear. The main drawback of those studies is that they are still at a prototype and proof of concept stage. Using sensors embedded in smartphones can be a better approach to activity recognition because smartphones are getting more and more common, they are likely to be with a user during his daily activities, have high processing power and adequate storage space, have relatively autonomy (before requiring a recharge), and are perceived as an unobtrusive device for most of the subjects.

To the best of our knowledge, there is no prior work that relies only on barometric pressure data coming from sensors embedded in smartphones for identifying individual's VDAs.

3. Rationale

Barometric pressure (or atmospheric pressure) is defined as the force per unit area exerted against a surface by the weight of the air above that surface. The standard unit for pressure is the pascal (Pa), which is equal to one Newton per square meter (N/m^2). In meteorology, the hectopascal (hPa) unit is mainly used; 1 hPa corresponds to 100 Pa.

Pressure depends on altitude h and temperature. For altitudes below 11 Km, their relationship can be defined as

$$h = h_0 + (T_0/k) * ((P/P_0)^{-(k*R/g*M)} - 1) \quad (1)$$

where P_0 and T_0 are the pressure and temperature at sea level (1013.25 hPa and 288.15° K), R is the universal gas constant (8.31432 Nm/ Kmol), k is the lapse rate/drop in temperature with altitude (0.0065° K/m) valid from sea level to 11 km, g is the standard acceleration due to gravity constant (9.80665 m/s^2), M is the molar mass of Earth's air (0.0289644 kg/mol) [1].

The function in Eq. (1) is non-linear, but continuous and monotonically decreasing with pressure P. Several factors can influence barometric pressure: atmospheric events, temperature and humidity changes, air conditioning/ventilation systems, window/door opening.

Atmospheric events comprehend meteorological conditions: when the weather changes from cloud to sunny or to rainy, pressure changes significantly. A cold wind can influence a reading and cause an error of around 10 m in the derivation of the altitude. However this class of events has a larger time scale than the sliding window of the activities we want to classify. The typical time duration of human activities we would like to classify is from 5 to 30 min, so sudden or slow changes in pressure due to modified weather conditions can be neglected.

Temperature changes can occur for example when a person exits from her/his warm office and enters a cold corridor and then goes outside a building; it can be shown that variations of 15°C result in an error of 14 cm in altitude estimation, while maximum error for a temperature span of 20°C is averagely of 20 cm in altitude estimation [35]. To compensate the error in pressure reading caused by sudden changes in temperature, current chips contain a temperature sensor bundled into the barometric sensor chipset. The driver reads both pressure as well as temperature, and compensates for the error in software. To verify the effects of temperature changes on pressure readings, we used a barometric pressure sensor enabled smartphone to measure pressure in a closed room of a building where an air conditioning system was operating. We subsequently exited the room and measured pressure outside. The results of the test showed that there were not any significant changes in the values of barometric pressure. We repeated the same experiment in a closed room where a heating system maintained the temperature at 26°C (while the temperature outside was 20°C). Also in this case, we didn't notice any significant pressure change.

When humidity changes, air density changes and therefore also pressure. However, when humidity increases from 50 to 90% the error introduced on altitude calculation is about 1 cm and therefore negligible [35]. In [36], authors found that when an air handler was operating, barometric pressure in a living room raised by 0.03 hPa. Similarly, a slight increase in air pressure of 0.015 hPa occurred in a master bedroom when its door was closed. Since these changes are below the relative pressure accuracy (0.1 hPa) of the sensors mounted in common smartphones, their impact on our measurements (and, indirectly, on our technique) is negligible.

If an application performs height estimation with barometric sensors, there is a need to constantly calibrate the values of base pressure and temperature, especially in outdoor conditions. This information is indeed provided by most airports and weather stations. Indoor height estimation can be done quite precisely but only with a constant update of reference values as it is shown in [37]. As illustrated in the following sections, we will use differential measurements to recognise activities, thus we can ignore reference values.

In real-life situations, it is highly unlikely that external factors could mimic human activities and create artifacts. For example, to mimic stair climbing the temperature should decrease regularly at around 20°C per second, to mimic riding an elevator (speed ca. 1.5 m/s) the opening of a door/window should bring an increase in pressure of 15–25 Pa per second. Furthermore, artifacts of small entity can be smoothed by using a proper smoothing algorithm as

[1] International Organization for Standardization (ISO), Standard Atmosphere,ISO 2533:1975, 1975.

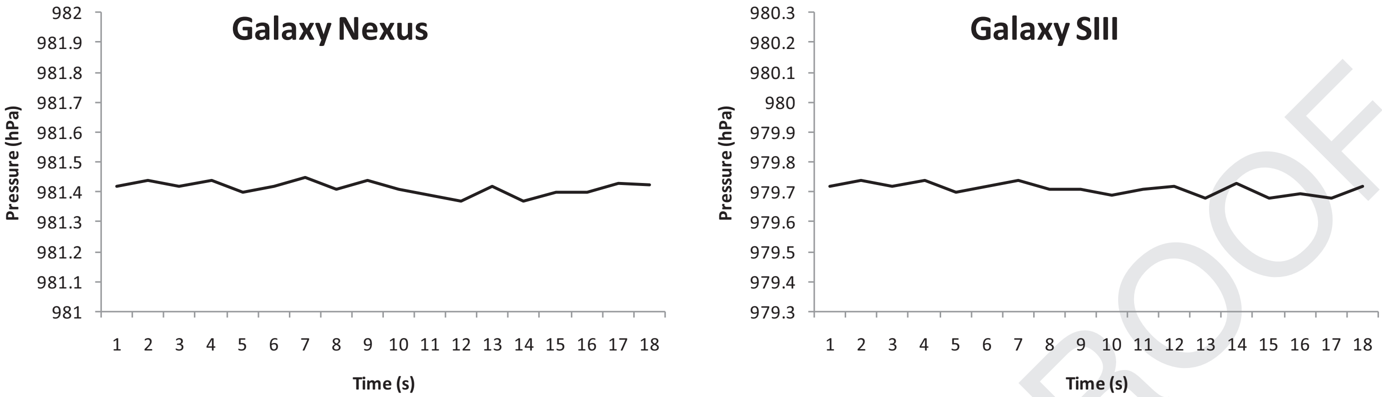


Fig. 1. Barometric pressure while a subject was standing still.

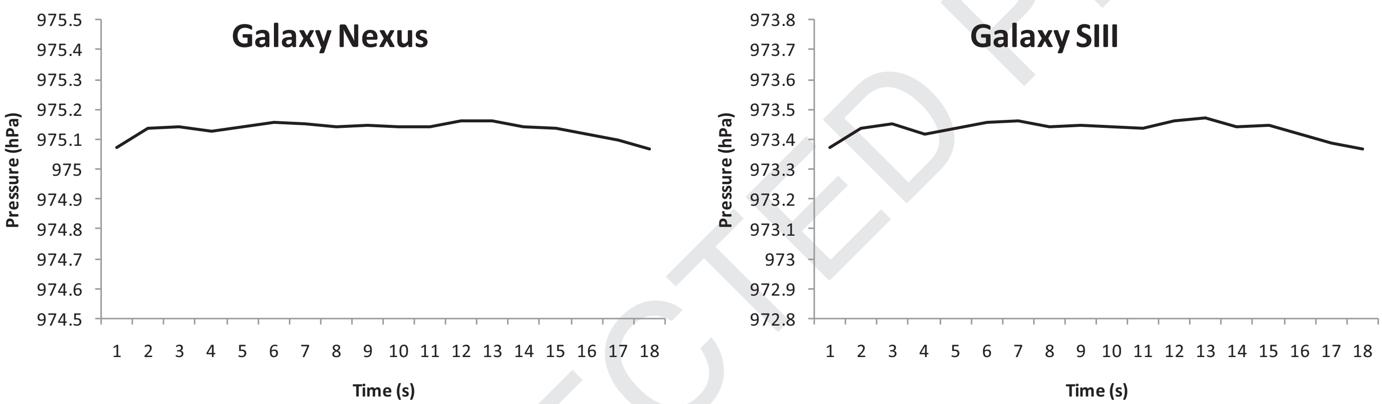


Fig. 2. Barometric pressure while a subject was walking on the same floor.

the one we used for our VDAs detection algorithm that is described in the following sections. In conclusion, Eq. (1) can be directly used for the present study without any modification.

4. System design

In this section, we describe the hardware (smartphones) used for our tests, the preliminary experiments we made to investigate the characteristics of barometric pressure in different scenarios, the process for collecting training data, and the method for deriving height information from pressure.

4.1. Hardware

To test the dynamics of pressure variations and check if they are the same for different sensors models, we conducted preliminary experiments using different mobile phones. For our tests, we used two Android v4 mobile phones models: Samsung Galaxy Nexus and Samsung Galaxy SIII. The first phone is equipped with a Bosch Sensortec BMP180 digital barometric pressure sensor, while the Samsung Galaxy SIII mounts a STMicroelectronics LPS331AP chip. Both sensors are based on piezo-resistive MEMS technology and have low power consumption (average current consumption in advanced mode is respectively, 30 and 32 μA). The relative accuracy for pressure is ± 0.12 hPa for the BMP180 and ± 0.1 hPa for the LPS331AP chip.

4.2. Characteristics of barometric pressure

We first analysed the trend of raw pressure readings.

Android does not allow to set the sampling time for sensors data. We determined empirically that using the "SENSOR_DELAY_FASTEST" rate for sensors events (0 μs data delay) yields the best results. After each sensor read, we waited one second prior to the next read operation, thus emulating a sampling time of roughly 1 Hz.

To mitigate the residual noise, we applied the double exponential smoothing method [38] to our time series because of its trend-tracking properties. If x_t is the raw data sequence of observations starting at time $t = 0$, s_t is used to represent the smoothed value at time t, and b_t is the best estimate of the trend at time t. Double Exponential Smoothing is given by:

$$\begin{aligned} s_t &= \alpha x_t + (1-\alpha)(s_{t-1} + b_{t-1}) \\ b_t &= \gamma(s_t - s_{t-1}) + (1-\gamma)b_{t-1} \end{aligned} \tag{2}$$

where α is the *data smoothing factor*, $0 \le \alpha \le 1$, and γ is the *trend smoothing factor*, $0 \le \gamma \le 1$. The initial values can be taken as $s_1 = x_0$ and $b_1 = x_1 - x_0$. The smoothing factor, for both the data and the corresponding trend, represents the importance applied to the most recent sample. We used 0.5 for both factors.

On other smartphone models (e.g., Nexus 5), the barometer chips performs smoothing internally, thus a smoothing technique in the code is not required [31].

To study the characteristics of pressure for the VDAs of different subjects, we recorded pressure in six different scenarios: while a subject was standing, while she was walking on the same floor of a building, while she was climbing and descending the stairs between different floors of a building, while she was riding an elevator from the bottom to the second floor of a building and, finally, while she was riding a cable-car. Results show that the pressure remains stable when a user is standing (Fig. 1) and is relatively stable when a user walks on the same floor (Fig. 2). When a subject is climbing/descending stairs the pressure decreases/increases.

Please cite this article as: S. Vanini et al., Using barometric pressure data to recognize vertical displacement activities on smartphones, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.02.011

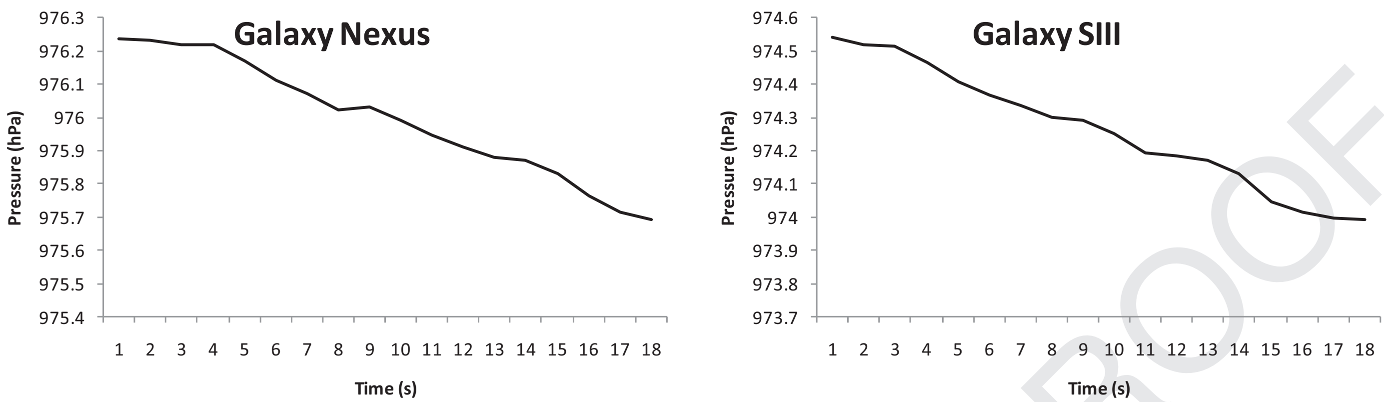


Fig. 3. Barometric pressure while a subject was climbing stairs.

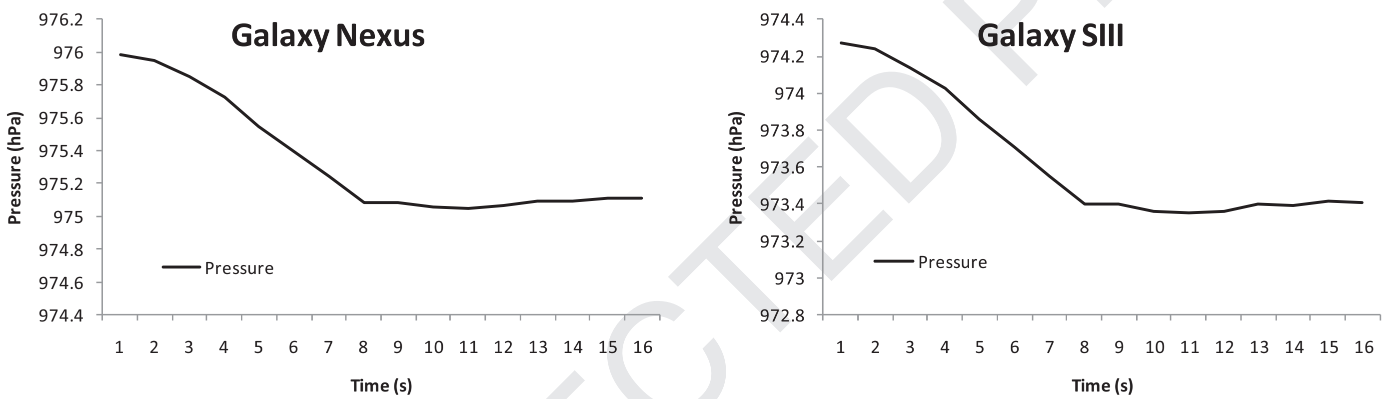


Fig. 4. Barometric pressure while a subject was riding an elevator up.

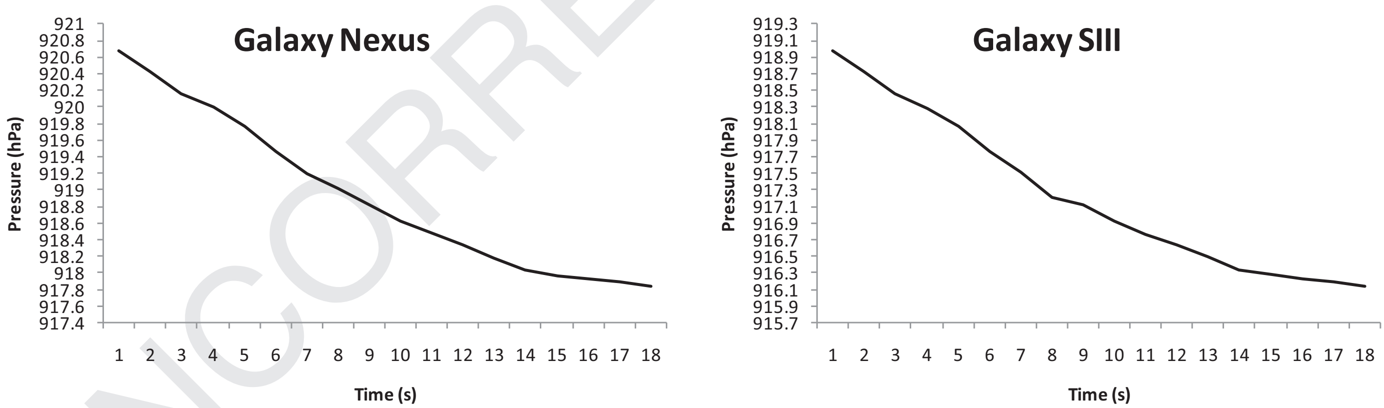


Fig. 5. Barometric pressure while a subject was riding a cable-car.

Fig. 3 shows the evolution of pressure while a subject was climbing stairs. When a subject is riding the elevator up, the pressure decreases at a higher rate than when climbing stairs, as illustrated in Fig. 4. Finally, when she rides a cable-car, barometric pressure varies significantly (Fig. 5). It can also be observed that the vertical speed of a user has the same dynamics – and can be easily inferred from the dynamics – of pressure variation over time. For example, the rate of pressure variation in elevators is higher, due to their higher rate of velocity.

The most important conclusion we derived from our set of experiments was that each VDA has distinct dynamics with regard to pressure variations. Furthermore, the standing and walking scenarios exhibit the same behaviour in terms of pressure variation. Finally, although absolute values for pressure were different, dynamics were similar for both phones, thus the pressure variation over time is independent of the type of sensor used for measuring it.

4.3. Training data collection

For our set of experiments in VDA recognition using barometric pressure data, we studied four types of activities:

- Standing/walking.
- Climbing stairs.
- Riding an elevator.
- Riding a cable-car.

The direction (up/down) while climbing stairs, riding an elevator and riding a cable-car can be automatically inferred by computing the difference between the starting pressure and the end pressure (the direction is "up" if the pressure decreases, "down" if it decreases)

To collect training data, we deployed an Android application that samples barometric pressure at a frequency of roughly 1 Hz (see Section 4.2). Subjects can label their activity by selecting the corresponding value from a spinner. Pressure values are recorded once a subject presses a *Start* button. Registration of pressure values is stopped after pressing a *Stop* button. Pressure data collected between the start and stop times are labelled with the name of the associated activities and stored in a SQLite database on the smartphone.

4.4. Feature computation

We performed features extraction on sliding windows with 50% overlap, which is fairly common in the literature [23,39,40]. We used a window of 8 samples (roughly 8 s). The features that we extracted from the sliding windows of barometric pressure were:

- Standard deviation.
- Slope: defined as the ratio between the change in barometric pressure over the window. This physical quantity gives an indication of how pressure varies over vertical displacement time.

5. Experiments and data analysis

We formulated activity detection as a classification problem, where classes are represented by labelled VDAs and test data instances are represented by the set of extracted features (standard deviation and slope) over barometric pressure measurements, collected using our Android application. We divided data sets into two different settings:

- Experiments with trained subjects (researchers).
- Experiments with untrained and unsupervised subjects.

We discriminated between the above two settings because the performance of classifiers may be significantly worse when applied on data collected by untrained persons, in real-world conditions. In the latter scenario, there are fewer constraints (e.g., subjects are not told exactly where and how to perform activities) with respect to a lab environment, where data is collected by scientists who perfectly know the behaviour of the phenomenon under observation. This observation has been remarked many times in related studies in literature. For example, [41] reported 95.8% activity recognition rates for data gathered in laboratory, but recognition rates dropped to 66.7% for data gathered outside the laboratory, in unconstrained settings.

Experiments were carried out by ten subjects: five from the academic community and five externals. Subjects performed their daily routines by recording and manually labelling actions corresponding to the four activities to detect, using our Android application. Training sets were acquired from the subjects themselves, in their workspaces, indoors and outdoors. On average, we gathered about 30 min of data per activity (around 1800 pressure readings per activity), per subject, except for the cable-car scenario, where traces were registered only by one trained and one untrained person, for about 15 min each. The phones we used for the experiments were the Samsung Galaxy Nexus and the Samsung Galaxy SIII. It is important to point out that experiments targeting the climbing stairs and riding elevator scenarios were performed in buildings with different layouts, without any prior knowledge of the building layout or any information about floor heights and about the number of steps between each floor (for the staircase scenario). In detail, floor height was in the range of (2.70; 3.40) m, while step height varied between 13 and 16 cm. Furthermore, experiments were run in different places, located at different elevations.

Table 1
Average values for features (standard deviation and slope) extracted from barometric pressure.

Activity	Std. Dev. (hPa)	Positive slope (hPa/s)	Negative slope (hPa/s)
Standing/Walking	0.00064	0.00456	−0.00438
Climbing stairs	0.006359	0.02671	−0.02961
Riding elevator	0.0828	0.1078	−0.0101
Riding cable-car	0.60346	0.2965	−0.29342

Table 2
Recognition accuracy for the activities studied with trained and untrained subjects for the J48 classifier.

Activity	Trained users (%)	Untrained users (%)
Standing/Walking	98.44	87.61
Climbing stairs	91.62	77.44
Riding elevator	95.14	88.25
Riding cable-car	99.99	99.99

Table 1 shows the average values for the features (standard deviation and slope) we extracted from the barometric pressure data, for the trained user scenario. We obtained similar results also for the untrained user scenario. As expected, the average value of the standard deviation for pressure increases as the amplitude of the vertical movement associated to the correspondent activity increases. The values for slope (i.e., the variation of pressure over time) confirm that pressure is almost stable when a subject is standing or walking, while it varies significantly when a subject is riding an elevator, and even more, when she/he is riding a cable-car. The dynamics for positive and negative variations are almost similar. Standard deviation and slope of pressure are correlated, since they both reflect a large amount of variation in the measurements of barometric pressure.

We split our analysis into two sections. In Section 5.1, we present machine learning algorithms, which are computationally cheap. A recurrent neural network algorithm – comparatively more computationally intensive – is instead presented in Section 5.2.

5.1. VDA detection with computationally cheap machine learning algorithms

We used the Weka Machine Learning Algorithm Toolkit [42] and evaluated the performance of two base-level classifiers: J48 Decision Trees and Naive Bayes. We found a quite significant difference between the trained and untrained scenario.

The highest recognition accuracy is reached by the J48 decision tree classifier. It was able to distinguish between the different activities with 95.06% average accuracy in the trained scenario and 83.20% average accuracy in the untrained scenario. In the trained scenario, 1.56% of the standing/walking instances were incorrectly classified as climbing stairs, while 5.13% instances of the climbing stairs scenario were wrongly detected as standing/walking and 3.25% as riding elevator. Finally, 4.86% instances of the riding elevator scenario were incorrectly detected as climbing stairs. Table 4 summarizes the aforementioned results. We found a similar behaviour also for the untrained scenario, but for the standing/walking case, where traces collected by untrained individuals were also classified as riding elevator (Table 5). Table 2 shows the results of activities recognition for the J48 classifier, while Table 3 shows the performance results for the Naive Bayes classifier. As it can be noticed, classification accuracy for the riding cable-car scenario is almost 100% in both cases: this is not surprising because this scenario has the lowest relative standard deviation of the training data (0.2 for standard deviation and 0.11 for slope).

Table 3
Recognition accuracy for the activities studied with trained and untrained subjects for the Naive Bayes classifier.

Activity	Trained users (%)	Untrained users (%)
Standing/Walking	94.91	90.38
Climbing stairs	88.78	68.1
Riding elevator	90.67	73.6
Riding cable-car	99.99	99.99

Table 4
Distribution of incorrectly classified activities for trained subjects with the J48 classifier.

Activity	Standing/ Walking	Climbing stairs	Riding elevator	Riding cable-car
Standing/Walking	–	1.56%	0%	0%
Climbing stairs	5.13%	–	3.25%	0%
Riding elevator	0%	4.86	–	0%
Riding cable-car	0%	0%	0%	–

Table 5
Distribution of incorrectly classified activities for untrained subjects with the J48 classifier.

Activity	Standing/ Walking	Climbing stairs	Riding elevator	Riding cable-car
Standing/Walking	–	11.49%	0.89%	0%
Climbing stairs	20.50%	–	1.57%	0%
Riding elevator	0%	11.75	–	0%
Riding cable-car	0%	0%	0%	–

The main reason the J48 classifier has better performance than the Naive Bayes classifier is that Naive Bayes makes use of all the features extracted, and analyses them individually as though they are equally important and independent of each other. This is not our case, because standard deviation and absolute value of the pressure slope are correlated (as explained in Section 5).

5.2. VDA detection with a recurrent neural network

To show the effectiveness of our barometric pressure-based approach, we tested a recurrent neural network to learn and classify our pressure time series composed of long time lags of unknown size between different activities. We used the PyBrain[2] machine learning library to implement a Long Short-Term Memory (LSTM) neural network. We chose LSTM because the activities to discriminate are repetitive (e.g., if a subject is climbing stairs at time t, it is highly likely that she will still be climbing stairs at time $t+1$) and they can be modelled with a recurrent network. Furthermore, LSTM outperforms other recurrent networks in many areas (e.g., regular, context-free and context sensitive languages [43]; handwriting recognition [44]; discriminative keyword spotting [45]). Another key factor is that LSTM can handle very large time lags, say of the order of several hundreds or thousands [46].

As any recurrent neural network, LSTM uses feedback connections to store representations of recent inputs events. In addition, it contains blocks that automatically determine when an input is significant enough to be stored. For training, we used the Rprop [47] supervised learning technique, which is an adaptive gradient based technique (computation of the gradient of an error measurement in weight space) known for its high convergence speed, accuracy and robustness.

With regard to the network architecture, we used 2 input units (standard deviation and slope) and 1 output unit (activity to be detected). For LSTM, we used 5 hidden units, an output layer with a softmax function (because we are doing classification), and a recurrent connection from the hidden to the hidden layer that looks one timestamp back in time. The input layer has connections to all units in the hidden layer. The output layer received connections only from the hidden layer. Since we used the Rprop trainer, all training samples have the same weight.

[2] http://www.pybrain.org

Table 6
Average test error rates with trained subjects for the LSTM neural network.

Activity	Test error rate(%)
Standing/Walking	1.29
Climbing stairs	0.22
Riding elevator	1.47
Riding cable-car	1.01

5.2.1. Results

We extrapolated 1500 samples from the set of measurements collected by the five trained users. The resulting training set was randomly split into 60% training and 40% test data sets. We initially ran some training iterations to set the values of the Rprop parameters that provided the minimum test error. After those tests, we decided to use 0.4 for *etaminus* (factor by which step width is decreased when overstepping) and 4 for *deltamax* (maximum step width). Then, we ran 50 training iterations, each of which stopped after the training module converged.

The LSTM algorithm nearly always learns to solve the VDA recognition task. The best test set error was only 0.22%. On average, the training module converged after 120 epochs. Table 6 shows the details of the test error rates for each activity.

5.3. Weather dependence

To demonstrate that our VDA detection algorithm is independent of changing weather conditions, we collected 24 h of traces with different weather conditions. We started from scattered clouds to variable clouds with isolated rain showers. The phone was lying still in the same position and we checked if changes in the barometric pressure under such variable weather conditions could trigger false positives and be detected as either climbing stairs, or riding an elevator or riding a cable-car.

The accuracy of the standing/walking mode was found to be 99.45%, showing that weather changes do not impact on the capability of our algorithm to detect VDAs. This is because weather drift occurs over a larger time scale than the temporal duration of our VDAs. Furthermore, rather than having frequent ups and downs associated with rapid pressure changes, weather drift is usually in one direction, and gradual.

Fig. 6 shows the pressure variation during a day due to changes of atmospheric events for a phone laying in the same position, along with the indication of the VDA detected. As it can be seen from the picture, the pressure trend follows weather changes. Furthermore, false positives (lighter lines in the picture) were triggered only relatively to the climbing stairs scenario.

5.4. Power consumption

We measured the power needed to retrieve barometric pressure values. Measurements were made at two different levels of granularity.

First, we implemented an App for monitoring significant changes in the battery level – specifically when a device enters a low battery state. The App can also be configured for reading

Please cite this article as: S. Vanini et al., Using barometric pressure data to recognize vertical displacement activities on smartphones, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.02.011

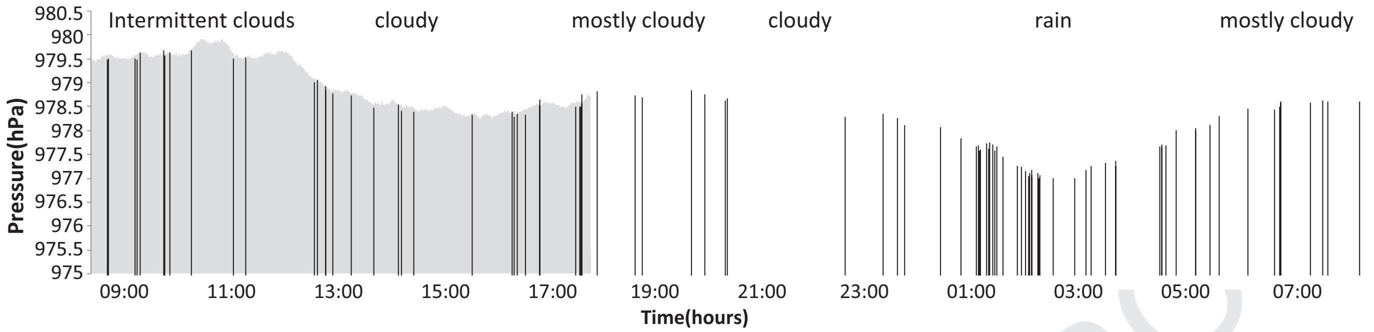


Fig. 6. Evolution of barometric pressure in a day along with the indication of false positives (darker lines) in the detected activity, while a phone was standing still.

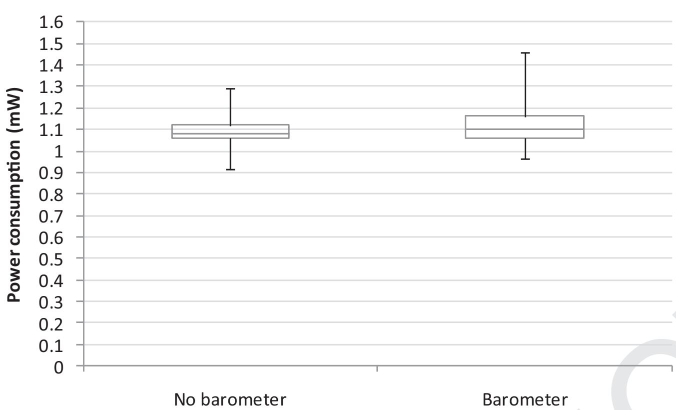


Fig. 7. Power consumption without and with (1 Hz) pressure readings.

barometric pressure data at a sample rate of approximately 1 Hz. It keeps the screen on at full brightness.

We started from a full-charged device, stopped all applications and services, run our application and measured the time until a low battery state was detected with pressure readings and without pressure readings. It took respectively 5.21 and 5.7 h to reach the low battery state, thus showing an overall impact well below 10%.

To get a more accurate measure of the power (in mW) needed to read values from pressure sensors, we employed a portable open sourced power monitor (POEM) [48] – similar to BattOr [49] – using the open source hardware platform Arduino [3]. To synchronize the external board with the smartphone, we used LED2LED communication [50] between a LED mounted on the Arduino board and the camera on the smartphone. POEM offers *mW* accuracy and a sampling rate down to *ms* for measuring power consumption.

Fig. 7 shows the boxplots of power consumption for two different scenarios: idle mode and pressure readings at a sample rate of approximately 1Hz. As it can be seen from the picture, the increase of power consumption due to the sampling of barometric pressure is negligible: the value of the second quartile for the “pressure reading” scenario is indeed slightly higher than the corresponding value in the “no pressure reading” scenario. The variability of power consumption is higher in the “pressure reading” scenario, where it can reach at most about 1.5 mW. This means that the power required to read pressure data is not always constant (this could be due to the OS). The average value of energy consumed by the smartphone when the App was reading pressure values from the barometric sensor was 1.23 mW, while it was 1.1 mW when the App was in idle state. Therefore, energy required for reading pressure values was about 0.13 mW on average. This value is in line with the current consumption (30μA) of the pressure sensor specified in the technical data sheet. In fact, since voltage supply required by the smartphone is 3.7 V, the power needed to read barometric pressure is about 0.12 mW. The remaining amount (0.01 mW) is due to the OS. It is important to emphasize that power consumption did not change when we moved (vertically) the device (which is the gesture that allows to discriminate about the different activities).

6. Comparison with other sensors and methodologies

As described in Section 2, the current approaches to VDA recognition using sensors are mainly based on accelerometers and GPS sensors. In the following sections, we compare the performance of our VDA detection based on pressure sensors with such approaches in terms of accuracy, energy efficiency, indoor effectiveness and phone position independence.

6.1. Accelerometer-based approaches

We start our comparison analysis with accelerometers. In terms of accuracy, Bao et al. [23] obtain the best performance results with decision tree classifiers, getting an accuracy rate of 95.67% for standing still, but recognition rates were significantly lower when riding elevators and climbing stairs (respectively 43.58 and 85.61%). Lester et al. [26] achieved their best result for climbing stairs, where the classification was correct 95% of the time. The accuracy for descending stairs and riding elevator up/down was respectively 89%, 87.3% and 84.6%. Authors were also able to recognize walking and standing activities, but in the latter accuracy was very low (55%). Results obtained by Kwapisz et al. in [21] show that the accuracy of recognizing the standing activity was up to 93.3%, while climbing stairs was inferred with about 60% accuracy at best. Krishnan et al. [22] show that Boosted Decision Stumps (Adaboost) classifiers have the best performance, achieving about 90% recognition accuracy for walking, standing, and climbing stairs scenarios. Khan et al. in [14] claim a 99% accuracy rate for the detection of sitting, 95% accuracy for walking and climbing stairs, and 92% accuracy for descending stairs.

Fig. 9 offers an overview of the accuracy rates in recognizing VDAs for the methods described above and our method (in the best case of trained users). Accuracy rates obtained with our barometric-based approach are in line with – and in some case better than – the numbers reported above.

With regard to energy efficiency, to get comparable data, we implemented an App that measures the acceleration force (in m/s^2) applied to a smartphone on all three physical axes (x, y, and z) and monitored the power consumption with our POEM tool. The App listens to accelerometer sensor events at a sampling rate of approximately 1Hz (as in the case of the App for reading

[3] http://www.arduino.cc

Please cite this article as: S. Vanini et al., Using barometric pressure data to recognize vertical displacement activities on smartphones, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.02.011

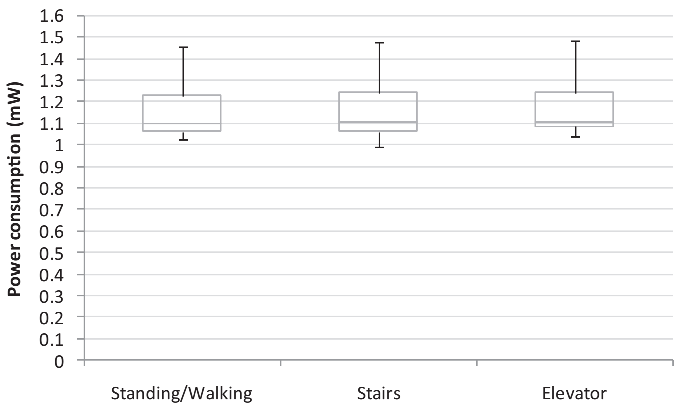


Fig. 8. Power consumption for reading acceleration data.

pressure data). We reproduced three of the four VDAs types studied in the paper. We omitted the cable-car scenario because accelerations related to this use-case are not easy to reproduce as they are very high. For example, during our set of experiments in VDA recognition, we measured an average acceleration of 2.6 m/s^2. Fig. 8 shows the boxplots of power needed for reading acceleration data in a temporal window of 10 s, in each of the three scenarios considered. We measured the power consumption at a sample rate of 100 ms. From the plots, it can be seen that the variability of the power consumption is almost the same in all three scenarios, but it is in any case higher than in the "no pressure reading" scenario described in Section 5.4. The power required to measure acceleration was overall the same in all three cases, so it does not depend on the intensity of the movement. On average, the power consumed by the App when reading acceleration data was about 1.37 mW: this means that energy required to read data from the acceleration sensor is about 0.27 mW.

The accuracy of activity recognition based on accelerometer data depends on the position of the sensors (or phone) and their orientation [14,26,51–53]. Conversely, during our experiments in VDA detection, we intentionally failed to control the position of the phone, as it happens in a daily usage pattern. We found that under this realistic conditions, the accuracy of VDA detection was unaffected by changes in the phone's on-body location (in pockets, hands, or even bags) and orientation of the phone. This finding is one of the main strengths of barometer pressure-based activity recognition and is also one of the key differences from accelerometer-based activity detection.

Table 7
Average power consumption in GPS receiver chips on smartphones.

Manufacturer	CSR	u-blox	MediaTek	Sony
Model	SirfSTAR IV GSD4t	Max-7 u-blox 7	MT3333	CXD5600GF
Continuous tracking (1 Hz)	40 mW	50 mW	19 mW	10 mW
Cyclic tracking (1 Hz)	8 mW	13.5 mW	N.A.	N.A.

6.2. GPS-based approaches

We have also compared barometers and GPS sensors in terms of activity recognition accuracy, height estimation, power consumption and sensor-position independence.

As already described in Section 2, methods based on GPS data for activity detection have low-level accuracy and flexibility.

Compared to GPS positioning, the main advantage of a pressure-based approach to vertical displacement measurement is that, as widely reported in literature, the accuracy of the barometer height estimation exceeds that of GPS. Furthermore, barometer is not subject to shadowing as GPS (which also impacts on the accuracy and availability of the altitude measurement), thus it can be used in an indoor environment.

In terms of power consumption, it is well-know that smartphone battery usage increases a lot when GPS interface is on [54]. Table 7 lists the average power consumption claimed by the most popular manufacturers (CSR, u-blox, MediaTek, Sony corp) of GPS receiver chips for smartphones. The table distinguishes among continuous tracking and cyclic tracking modes. In the first mode, the receiver continuously tracks all the available satellites to achieve the best possible position accuracy. In cyclic tracking, the receiver employs intermittent tracking to conserve power: a significant amount of power is saved by periodically turning off the Radio Frequency (RF) front-end and most of the hardware in this mode. From the table, it can be seen that even the lowest values for power consumption are higher than the average power (0.13 mW) consumed by the pressure sensor.

Finally, since the GPS accuracy is very low, the position of the phone does not have a significant impact on activity detection.

Table 8 recaps the features compared in our analysis for the three sensors, using star ratings (out of a maximum of 3 stars). It can be noticed that GPS is the only sensor that is not working indoor, thus it cannot be used neither for activity detection nor for

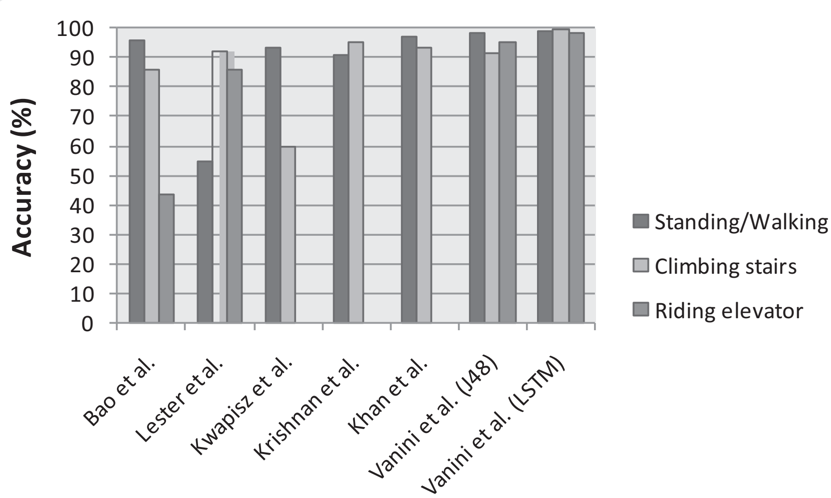
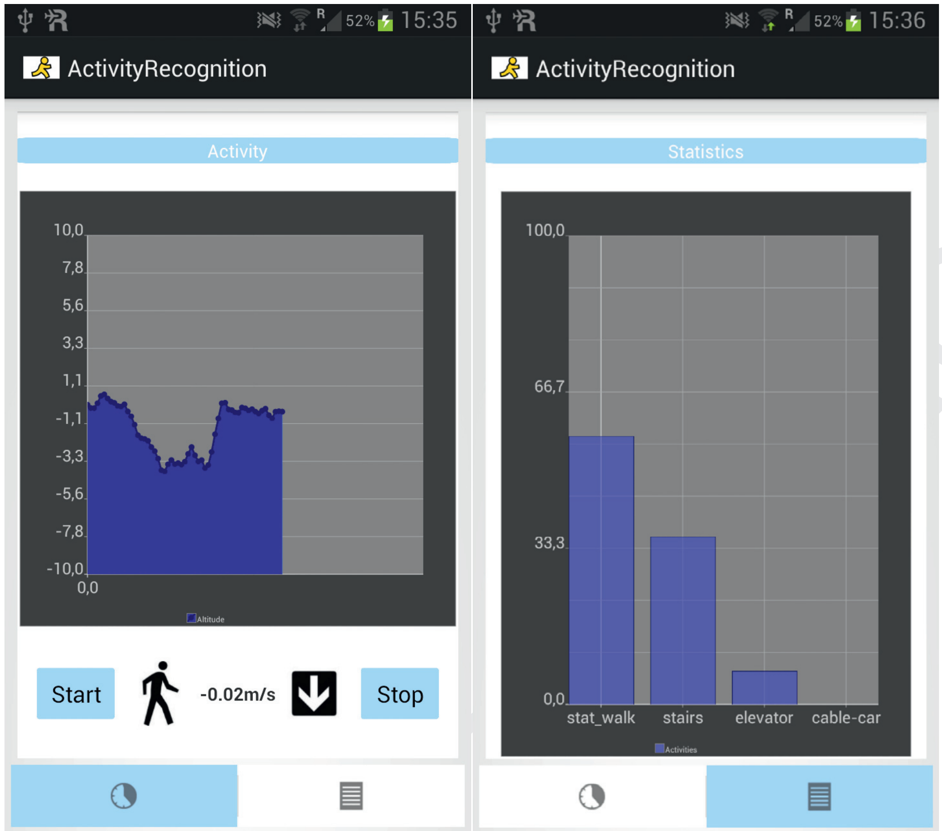


Fig. 9. Comparison of accuracy rates for inferring VDAs with accelerometers.



(a) Detected activity and elevation profile (b) Occurrences (%) of detected activities

Fig. 10. GUI of the App implementing activity recognition.

Table 8
Comparison of sensors for VDA detection on a 3 star bases.

Sensor	Detection accuracy	Energy efficiency	Indoor effectiveness	Position independence
Accelerometer	***	**	***	*
GPS	*	*	N.A.	**
Barometer	***	***	***	***

altitude estimation in that scenario. Barometer is the only sensor that has a 3 star rating for all the metrics analysed.

To summarize, the performance of our proposed pressure-based method is comparable and in most cases better than the performance of traditional systems based on acceleration data, but it has the significant advantage of being energy-efficient.

7. Use case scenario: an application for detecting user activities and measuring altitude differences

As a use case scenario, we implemented an App for Android to infer the VDAs carried out by a user and to show statistics about them.

We used the Weka library for Android[4] and utilized a training set of instances that we collected during our experiments. Although LSTM has better accuracy, its power consumption is significant, as reported in [55]. To comply with energetic constraints, we chose the J48 decision tree classifier. In fact, J48 provides very good accuracy results and has a linear complexity for both training and inferencing activities, resulting in a small impact on energy consumption.

The App also estimates and shows the elevation profile during a user's journey. This is done by converting the pressure measured by the barometer to elevation information, using a simplified version of Eq. (1). As it can be seen from that equation, absolute height information cannot be calculated without the proper knowledge of local sea level pressure, which varies depending on weather conditions. Reference barometer information can be obtained via an auxiliary TCP/IP server connection, which is not always available. Furthermore, a high level of accuracy for altitude is not required. For these reasons, we can use a simplified version of Eq. (1). Assuming constant weather conditions and that the typical air pressure at the sea level is 1013 hPa, it can be easily derived that near the Earth's surface a difference of 1hPa corresponds approximately to 8.4 m in elevation.

Finally, the information about elevation changes is also used to derive the vertical speed of a user.

This use case scenario shows the potential of pressure sensors and the large number of application fields where pressure sensors can be employed.

Fig. 10 shows the user interface of the App, which is composed of two tabs. The home tab contains two buttons for starting and stopping activity detection. Elevation changes are shown in a graph, while current activity and direction are shown as an icon. Vertical speed (in m/s) is displayed at the centre of the screen. The second tab contains a graph that shows statistics about the activities performed during a start-stop session and their occurrences.

8. Conclusion

The barometer is one of the least frequently used sensors on smartphones, but is also one of the most promising. In this work, we demonstrated the advantages of the use of barometric pressure

[4] https://github.com/rjmarsan/Weka-for-Android

data from sensors embedded in mobile phones to recognize vertical displacement activities. We evaluated the performance of three different models to infer user activities and found that the LSTM recurrent neural network has a very high accuracy rate. However, J48 decision tree algorithm is a good choice for resource-constrained devices owing to its fairly high accuracy, its low computational overhead, and (consequently) its low energy consumption. We implemented an Android application that integrates the J48 decision tree algorithm and infers the activity performed by a user. The application uses barometric pressure data to provide information on the vertical distance travelled by a user and also shows it instantaneously on a graph. We also showed that barometric pressure sensors have many advantages over sensors traditionally used for activity recognition (namely, accelerometers and GPS) in terms of accuracy, energy efficiency, indoor effectiveness and independence from the phone position. The use of barometric data for activity recognition is very advantageous, as many applications that cannot be correctly recognized with accelerometer can be easily inferred with pressure data. Furthermore, the barometric sensor can enhance the quality of accelerometer sampling whenever a vertical displacement is present but is not the main movement. Finally, the barometer can also be used as a trigger to accelerometer sensing when the barometer itself cannot achieve a sufficient quality, resulting in a more power-efficient approach.

Future work includes the use of multiple sensors for activity detection and the implementation of a mechanism for switching between sensors – and their underlying methodology – depending on geo-position, predominant activity, and objective functions like e.g., battery life and accuracy optimization.

References

[1] S. Giordano, D. Puccinelli, When sensing goes pervasive, Pervasive Mobile Comput. 17 (2015) 175–183.
[2] A. Puiatti, S. Mudda, S. Giordano, O. Mayora, Smartphone-centred wearable sensors network for monitoring patients with bipolar disorder, in: Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, IEEE, 2011, pp. 3644–3647.
[3] P. Kotzanikolaou, C. Patsakis, E. Magkos, M. Korakakis, Lightweight private proximity testing for geospatial social networks, Comput. Commun. 73 (2016) 263–270.
[4] E. Miluzzo, M. Papandrea, N.D. Lane, A.M. Sarroff, S. Giordano, A.T. Campbell, Tapping into the vibe of the city using vibn, a continuous sensing application for smartphones, in: Proceedings of 1st International Symposium on From Digital Footprints to Social and Community Intelligence, ACM, 2011, pp. 13–18.
[5] Z. Yu, Y. Liang, B. Guo, X. Zhou, H. Ni, Facilitating medication adherence in elderly care using ubiquitous sensors and mobile social networks, Comput. Commun. 65 (2015) 1–9.
[6] F. Theoleyre, T. Watteyne, G. Bianchi, G. Tuna, V.C. Gungor, A.-C. Pang, Networking and communications for smart cities special issue editorial., Comput. Commun. 58 (2015) 1–3.
[7] C.N. Scanaill, S. Carew, P. Barralon, N. Noury, D. Lyons, G.M. Lyons, A review of approaches to mobility telemonitoring of the elderly in their living environment, Ann. Biomed. Eng. 34 (4) (2006) 547–563.
[8] L. Panini, R. Cucchiara, A machine learning approach for human posture detection in domotics applications, in: Proceedings of the 12th International Conference on Image Analysis and Processing, 2003, IEEE, 2003, pp. 103–108.
[9] M. Papandrea, M. Zignani, S. Gaito, S. Giordano Cremonese, G.P. Rossi, How Many Places Around You? (2013).
[10] M. Zignani, M. Papandrea, S. Gaito, S. Giordano, G.P. Rossi, On the key features in human mobility: relevance, time and distance, in: Proceedings of 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, 2014, pp. 260–265.
[11] S. Vanini, S. Giordano, Adaptive context-agnostic floor transition detection on smart mobile devices, in: Proceedings of 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, 2013, pp. 2–7.
[12] M. Conti, C. Boldrini, S.S. Kanhere, E. Mingozzi, E. Pagani, P.M. Ruiz, M. Younis, From manet to people-centric networking: milestones and open research challenges, Comput. Commun. 71 (2015) 1–21.
[13] S. Giordano, D. Puccinelli, The human element as the key enabler of pervasiveness, in: Proceedings of 2011 the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), IEEE, 2011, pp. 150–156.
[14] A.M. Khan, Y.-K. Lee, S. Lee, T.-S. Kim, Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis, in: Proceedings of 2010 5th International Conference on Future Information Technology (FutureTech), IEEE, 2010, pp. 1–6.
[15] Y. Man, E.C.-H. Ngai, Energy-efficient automatic location-triggered applications on smartphones, Comput. Commun. 50 (2014) 29–40.
[16] B. Priyantha, D. Lymberopoulos, J. Liu, Littlerock: enabling energy-efficient continuous sensing on mobile phones, Pervasive Comput. IEEE 10 (2) (2011) 12–15.
[17] R.K. Ganti, S. Srinivasan, A. Gacic, Multisensor fusion in smartphones for lifestyle monitoring, in: Proceedings of 2010 International Conference on Body Sensor Networks (BSN), IEEE, 2010, pp. 36–43.
[18] A.M. Khan, A. Tufail, A.M. Khattak, T.H. Laine, Activity recognition on smartphones via sensor-fusion and kda-based svms, Int. J. Distrib. Sens. Netw. 2014 (2014).
[19] G. Liu, M. Iwai, Y. Tobe, D. Matekenya, K.M.A. Hossain, M. Ito, K. Sezaki, Beyond horizontal location context: measuring elevation using smartphone's barometer, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, pp. 459–468.
[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[21] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explor. Newslett. 12 (2) (2011) 74–82.
[22] N.C. Krishnan, S. Panchanathan, Analysis of low resolution accelerometer data for continuous human activity recognition, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, IEEE, 2008, pp. 3337–3340.
[23] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: Pervasive Computing, Springer, 2004, pp. 1–17.
[24] S. Boughorbel, J. Breebaart, F. Bruekers, I. Flinsenberg, W. ten Kate, Child-activity recognition from multi-sensor data, in: Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research, ACM, 2010, p. 38.
[25] A. El Halabi, H. Artail, Integrating pressure and accelerometer sensing for improved activity recognition on smartphones, in: Proceedings of 2013 Third International Conference on Communications and Information Technology (ICCIT), IEEE, 2013, pp. 121–125.
[26] J. Lester, T. Choudhury, G. Borriello, A practical approach to recognizing physical activities, in: Pervasive Computing, Springer, 2006, pp. 1–16.
[27] M. Elhoushi, J. Georgy, A. Wahdan, M. Korenberg, A. Noureldin, Using portable device sensors to recognize height changing modes of motion, in: Proceedings of 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, 2014, pp. 477–481.
[28] L. Liao, D. Fox, H. Kautz, Hierarchical conditional random fields for GPS-based activity recognition, in: Robotics Research, Springer, 2007, pp. 487–506.
[29] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001).
[30] L. Liao, D. Fox, H. Kautz, Blocation-based activity recognition using relational Markov networks,[in proc. 19th int, in: Proceedings of 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005, pp. 773–778.
[31] K. Sankaran, M. Zhu, X.F. Guo, A.L. Ananda, M.C. Chan, L.-S. Peh, Using mobile phone barometer for low-power transportation context detection, in: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, ACM, 2014, pp. 191–205.
[32] K. Komeda, M. Mochizuki, N. Nishiko, User activity recognition method based on atmospheric pressure sensing, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 2014, pp. 737–746.
[33] C. Mattmann, O. Amft, H. Harms, G. Tröster, F. Clemens, Recognizing upper body postures using textile strain sensors, in: Proceedings of the 11th IEEE International Symposium on Wearable Computers, 2007, IEEE, 2007, pp. 29–36.
[34] J. Cheng, O. Amft, P. Lukowicz, Active capacitive sensing: exploring a new wearable sensing modality for activity recognition, in: Pervasive Computing, Springer, 2010, pp. 319–336.
[35] C. Bollmeyer, T. Esemann, H. Gehring, H. Hellbruck, Precise indoor altitude estimation based on differential barometric sensing for wireless medical applications, in: Proceedings of 2013 IEEE International Conference on Body Sensor Networks (BSN), IEEE, 2013, pp. 1–6.
[36] J. Lstiburek, K. Pressnail, J. Timusk, Air pressure and building envelopes, J. Build. Phys. 26 (1) (2002) 53–91.
[37] B. Li, B. Harvey, T. Gallagher, Using barometers to determine the height for indoor positioning, in: Proceedings of the 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2013, pp. 1–7.
[38] C. Croarkin, P. Tobias, Nist/sematech e-handbook of statistical methods, NIST/SEMATECH (2006). July. Available online: http://www.itl.nist.gov/div898/handbook
[39] R.W. DeVaul, S. Dunn, Real-time motion classification for wearable computing applications (2001). 2001 Project Paper.
[40] K. Van Laerhoven, O. Cakmakci, What shall we teach our pants? in: Proceedings of the Fourth International Symposium on Wearable Computers, IEEE, 2000, pp. 77–83.
[41] F. Foerster, M. Smeja, J. Fahrenberg, Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring, Comput. Human Behav. 15 (5) (1999) 571–583.
[42] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.
[43] F.A. Gers, J. Schmidhuber, Lstm recurrent networks learn simple context-free and context-sensitive languages, IEEE Trans. Neural Netw. 12 (6) (2001) 1333–1340.

Please cite this article as: S. Vanini et al., Using barometric pressure data to recognize vertical displacement activities on smartphones, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.02.011

[44] A. Graves, J. Schmidhuber, Offline handwriting recognition with multidimensional recurrent neural networks, in: Advances in Neural Information Processing Systems, 2009, pp. 545–552.
[45] S. Fernández, A. Graves, J. Schmidhuber, An application of recurrent neural networks to discriminative keyword spotting, in: Proceedings of Artificial Neural Networks–ICANN 2007, Springer, 2007, pp. 220–229.
[46] A.F. Atiya, A.G. Parlos, New results on recurrent network training: unifying the algorithms and accelerating convergence, IEEE Trans. Neural Netw. 11 (3) (2000) 697–709.
[47] C. Igel, M. Hüsken, Improving the rprop learning algorithm, in: Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), 2000, Citeseer, 2000, pp. 115–121.
[48] A. Ferrari, D. Gallucci, D. Puccinelli, S. Giordano, Detecting energy leaks in android app with poem, in: Proceedings of 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE, 2015, pp. 421–426.
[49] A. Schulman, T. Schmid, P. Dutta, N. Spring, Demo: phone power monitoring with battor, in: Proceedings of the 17th ACM International Conference on Mobile Computing and Networking, MobiCom, vol. 11, Citeseer, 2011.
[50] S. Schmid, G. Corbellini, S. Mangold, T.R. Gross, An led-to-led visible light communication system with software-based synchronization, in: Proceedings of 2012 IEEE Globecom Workshops (GC Wkshps), IEEE, 2012, pp. 1264–1268.
[51] L. Sun, D. Zhang, B. Li, B. Guo, S. Li, Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations, in: Ubiquitous Intelligence and Computing, Springer, 2010, pp. 548–562.
[52] U. Maurer, A. Smailagic, D.P. Siewiorek, M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in: Proceedings of International Workshop on Wearable and Implantable Body Sensor Networks, 2006. BSN 2006, IEEE, 2006, pp. 4–pp.
[53] L. Atallah, B. Lo, R. King, G.-Z. Yang, Sensor placement for activity detection using wearable accelerometers, in: Proceedings of 2010 International Conference on Body Sensor Networks, IEEE, 2010, pp. 24–29.
[54] Y. Qi, C. Yu, Y.-J. Suh, S.Y. Jang, Gps tethering for energy conservation 1, in: Proceedings of 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015, pp. 1320–1325.
[55] A. Ferrari, D. Puccinelli, S. Giordano, Gesture-based soft authentication, in: Proceedings of 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2015, pp. 771–777.