Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

ARTICLE TYPE

Wearable Sensor Networks: A Measurement Study

Armir Bujari | Ombretta Gaggi* | Claudio E. Palazzi | Giacomo Quadrio

[1]Department of Mathematics ”Tullio Levi-Civita”, University of Padua, Padua, Italy

Correspondence
*Ombretta Gaggi. Email: gaggi@math.unipd.it

Abstract

Wearable technology is no longer science fiction. Thanks to the growing capability in the production chain to miniaturize complex electronics, a wide variety of gadgets that can be worn or included in dresses and accessories have emerged. These smart gadgets can collect data about the physical condition of the user and/or the environment providing the basis for innovative and valuable services. The main goal of this paper is to assess this context through field experiments undertaken in a testbed comprised of sensing hardware deployed on open source boards such as Arduino. Moreover, coupled with the sensing tier, we propose a proof-of-concept deployment architecture enabling a wide range of wearable sensors to collect and transmit data to a logically centralized unit.

KEYWORDS:
Wearables, Internet of Things, System design, Measurements

1 | INTRODUCTION

Even if wearable technology is not new, the first wearable computer can be traced back to 1955 [1], wearable devices have been relegated to science fiction for many years. Recently, the advent and market penetration of smart watches, activity trackers, open hardware platforms such as Arduino and Raspeberry Pi, equipped with communication and sensing capabilities have revolutionized our capability to collect and process data [2]. Sensing technology has been transparently included in ornaments, clothing, human tissue, etc., providing the basis for an autonomous, specialized *Wearable Sensor Network* (WSN) which, if exploited as a whole, could bring benefits weighting more than the sum of its parts [3,4].

It is a fact that nowadays wearable technology is pervasive in our lives: smart watches, glasses and bracelets offer a lot of useful functionalities and can collect information about the user (e.g., number of steps, heart rate, level of stress, position etc.) and the environment (e.g., sound, light, altitude etc.). Moreover, users themselves often contribute to this data collection effort for a common good, e.g., to provide information about road accessibility [5,6,7]. For all these reasons, Ang *et al.* coined the term of “people as sensors” since the user has become a source of data and fruitful information [8].

Wearable technology has found applications in many fields. Historically, its first applications are to be found in the military domain where wearables have been employed to enhance soldiers’ capabilities and increase safety [9]. Other important application fields are *fitness* & *wellness* [10] and rehabilitation [11] where wearable devices can be used to monitor user performance in time, and if necessary, intervene to correct harmful behavior, e.g., sedentary lifestyle. Wearable devices are also used to monitor elderly behavior for safety purposes [12,13].

Although many of these goals can be achieved even through smartphones, wearables are usually better accepted by the user since they are less invasive. As an example, consider a location control sensor for patients affected by Alzheimer disease that can be disguised as an ornament. Since wearable technology is becoming less and less expensive, wearable sensors are widespread with the potential of generating a huge amount of data. This data must be collected and further processed in order to be useful. Indeed, problems such as power management, efficient data gathering, storing and processing are becoming more and more important, i.e., even the best wearable object is useless if it drains its battery in a very short time.

In this paper, we present an analysis of a wearable sensor network testbed encompassing a wide range of sensing hardware embedded on open source boards such as Arduino. We discuss the design criteria involved and present real measurements discussing the tradeoffs that emerge. In addition, we present a conceptual framework, modeling a wearable sensor network comprised of a multitude of sensors used to acquire data, and more complex nodes that have also processing capabilities and/or interact with the user (e.g., a smartphone). The proposed framework is simple and flexible and can adapt to different scenarios. For instance, data can be transmitted through wireless communication (e.g., Wi-Fi and/or Bluetooth) and can be collected anonymously or with the user's identity. Furthermore, nodes can be easily added or removed as pluggable network components, providing functionalities on a per-need basis.

While preserving the general aspect of our study and without loss of generality, we focus on two specific case studies: a beach resort and a winter sports resort. Both these scenarios employ wearables as means to collect environment and user data in order to provide contextualized, personalized services. Our study also considers feasibility problems like sensor calibration and battery lifetime.

The paper is organized as follows: in Section 2 we provide some background information, discussing related works similar in spirit to ours. Section 3 states the problem we aim to investigate and some relevant case studies used as a means to better comprehend the design choices. Next, in Section 4 we discuss our envisioned system architecture, while in Section 5 we present the field tests performed with real sensors and discuss their outcome. Finally, in Section 6 conclusions are drawn.

2 | RELATED WORK

From a technological standpoint, the pervasive coverage of mobile sensing technology and the ubiquity of network connectivity are key factors that have boosted our capability to acquire and monitor different phenomena. Gained information can be related to physiological data (transparently) acquired from the user and/or environmental data related to the context the user is immersed in. The people-centric sensing paradigm has been around for some years and a significant research effort has been devoted to its development, ranging from algorithms and techniques proposed to measure specific environmental properties to new architectural and communication paradigms [14,15,16,17].

A testimony of this ever growing interest is the development of the fog/edge computing paradigm, advocating for some intelligence in the access network, sharing part of the computational burden with more capable, network provisioned servers [18,19,20]. Broadly speaking, the Internet of Things has, by some time now, been the focus of standardization efforts such as those undertaken in the 5G body of work [21,22].

With an emphasis on the sensing layer, commercial products have proven successful in the market and this pace of growth is expected to continue in the future. Wrist bands and smartwatches are amongst the most popular appliances in this category, having the capability to monitor and collect historical data about a user's physiological conditions and/or the environment. A lot of research work has been conducted in this domain, involving proof-of-concept systems built by exploiting open platforms such as Arduino and Raspberry Pi boards. Mentioning few representative examples, Jutila et al. propose a proof-of-concept wearable sensor vest with integrated wireless charging, designed to enhance the security of children [23]. The developed sensor vest provides information about the location and well-being of children, based on received signal strength indication, global positioning system, accelerometer, and temperature sensors. The authors in [24] propose a wearable training system that can be employed to facilitate the learning process of proper movement patterns in sports training. The system is comprised of a gesture-based user interface and a real-time biofeedback system consisting of one or more body-attached motion sensors, a processing device and a biofeedback device that are interconnected through a wireless medium.

Harnessing the power of crowdsourcing, in [25] the authors describe an application that determines pollution exposure indexes for people carrying mobile devices. Similar, Tse and Pau investigated the possibility to exploit user accessories embedded with sensing technology to monitor air quality, UV index and pollen [26]. A microblogging service is discussed in [27]; it uses mobile devices to record multimedia content in-situ and shares this content in real-time. Aram *et al.* propose a system for data acquisition using smartphones and specialized sensors [28]. In particular, they show how it is possible to acquire temperature and humidity values using low cost sensors and how these data could be forwarded through Bluetooth connectivity to a smartphone. In this way, it is possible to monitor conditions about a room or a particular environment, and alert when specific unhealthy conditions are reached.

PRISM [29] is a yet another framework that supports the participatory data acquisition of environmental data using off-the-shelf mobile device. This framework provides both an infrastructural component that orchestrates the mobile nodes participating in the data acquisition process, and a mobile component to be used with the smartphones that automatically collects data from the environment.

3 | PROBLEM STATEMENT

The main goal of this paper is to study and discuss the use of heterogeneous set of wearable sensors and to asses them in real life through field experiments. To manage wireless communication and data storage, we propose a proof-of-concept deployment architecture, enabling a wide range of wearable sensors to collect and transmit data to a logically centralized unit.

Moreover, to study the feasibility of the architecture, we present a measurement study undertaken with real sensing devices, built on top of the Arduino platform. The main motivation behind this study is the need for measurements related to the accuracy and energy expenditure of the adopted technology, undertaken in different scenarios. To the best of our knowledge, no other work in literature presents an empirical analysis like the one reported in this work.

To better understand the domain under scrutiny, we discuss two representative case studies, through which we show some possible uses of wearable sensors in everyday life. More in detail, we consider two scenarios where data are collected about the environment (e.g., UV/humidity) and about the user (e.g., position).

3.1 | Beach Resort

The first use case specializes the use of a WSN in a beach resort scenario. The idea is that each customer receives a set of gadgets that contain sensors used to monitor and collect data about the surrounding environment. As an example, the kit can contain hats equipped with GPS sensors to monitor children's position as well as a photovoltaic battery as its supply. Bracelets for the other members of the family can be equipped with UV sensors to prevent sunburns, T-shirts can contain humidity or temperature sensors and beach bags can seamlessly embed an Arduino board with communication, storage and processing capabilities. In this context, a client application (e.g., a smartphone app) is capable of accessing the collected information and/or retrieve contextualized, personalized information form the WSN system. The application can be downloaded on the fly from a dedicated pool of resources made available by the resort [30].

Information about UV, humidity, temperature, pressure and the position of the user(s) is recorded with the goal of providing services hosted in logically centralized server located at the seaside resort. The server receives the data from the WSN and upon processing can offer a wide range of personalized, contextualized services such as warnings about the weather forecast (e.g., to prevent sunburns), warnings in case a child is walking too far away, but also promotions and advertisements related to the beach resort. The collected data can be used also by a weather forecast station and/or by the beach resort itself.

3.2 | Winter Sports Resort

The second use case is similar to the first one, but pertains to a winter sports resort. Even in this case, the resort offers to its customers a sport kit which contains a set of gadgets equipped with sensing technology used to monitor and collect data about the surrounding environment. The collected data is used to create new services for customers but also to profile them and to obtain other information, such as weather forecast or statistics about the use of the ski lifts.

In this case, the kit can be composed by ski helmets, ski masks, caps, gloves, bracelets, backpacks, etc. The data collected is generally the same as in the previous case study (e.g., UV, humidity, temperature, pressure and GPS position of the users); they are used to provide augmented and personalized services thanks the processing capabilities of logically centralized processing units located in the winter sports resort. The data are sent from the WSN to a server that processes them in order to offer services such as information and warnings about the weather, advertisements, etc.

In this context, a user tracking service is not only useful for monitoring the children: it can be used to provide information about points of interest in the nearby (ski lifts, tracks, scenic tours, etc.), but also exploited in case of an emergency (e.g., in case of accidents) in order to track down the user. In this scenario, a remotely controlled drone with a video feedback can intervene aided by the WSN, helping rescue teams in finding the missing person.

4 | SYSTEM ARCHITECTURE

In this section, we present a framework for the envisioned WSN along with an accurate description of the reference architecture, the nodes comprising it and their capabilities. The main objective is to provide a *simple* and *flexible* framework, two fundamental qualities that allow us to instantiate the framework to different scenarios with similar requirements.

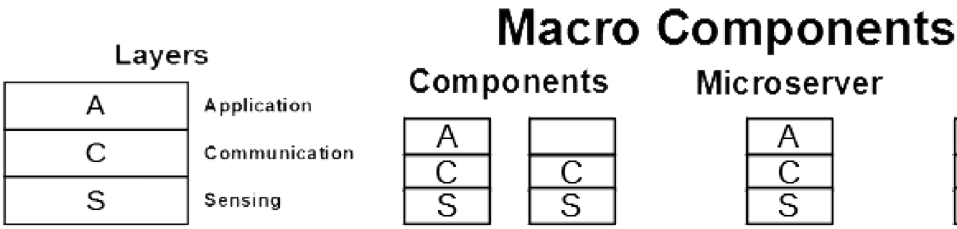


FIGURE 1 Framework layers

A WSN is composed by a set of *nodes* that cooperate to collect and process sensed data from the environment and/or the users. In the depicted scenarios, each sensing and processing device represents a system node and nodes may differ in their capabilities. Indeed, nodes can be grouped into three different categories, namely: *components*, *microservers* and *servers*.

4.1 | Node Architecture

To better describe the nodes involved in the system and their capabilities, we differentiate them based on the functional layers they are equipped with and operate. These layers are shown in Figure 1 and comprise the *Sensing layer*, the *Communication layer* and the *Application layer*. Each node can operate in all or in some of these layers, according to its characteristics. To this end, we assume the existence of layered WSN middleware with well-defined reference points. The nodes in the system implement some or all the functionalities by relying on specific APIs.

The *Sensing layer* is the lower layer and deals with data acquisition (i.e., *sensing*). According to the hardware equipment of each particular node, this layer is responsible for recording (and temporarily storing) data acquired by the sensors of that particular node. For instance, a T-shirt or a bracelet equipped with a UV sensor resides at the sensing layer.

The *Communication layer* is responsible for sending data through the network from the source to a destination. All nodes that are part of the WSN must have this layer enabled.

The *Application layer* instead manages the data at the destination, relying on a particular function running on a system node. This layer has storage, processing and, eventually, visualization capabilities. Only more complex nodes, like the server or the smartphone, reside in this layer.

Components are all those nodes that have the ability to collect and/or visualize data. They are able to work in all the layers of the framework, but their application layer is very simple since they are usually used as a user interface to visualize data and/or to collect feedback. Components are not involved in the actual computational effort of the system as they usually have limited computing capability. A wearable microcontroller equipped with a UV sensor and/or a humidity sensor falls in this category as well as a GPS tracker. These devices usually have low computing and storage capabilities and considering the wireless nature of our system this also implies low battery lifetime and reduced duty cycling. The components can operate in all the three layers or only in the Communication and Sensing layers (e.g., intelligent sensors). Smartphones are examples of components operating in all the three layers because they can be used both for data collection through their sensors, but also help visualise them, e.g., weather forecast.

Microservers are nodes that collect data generated by the components and perform some preliminary computation on them (e.g., data aggregation, summation). In our WSN, we adopt the concept of *Edge Computing* expressed in [31], envisaging computing services in the path between a logically centralized, networked server(s) and the end-users. This is generally due to the high amount of data generated by the involved devices, which can be temporary stored and processed at the network edge instead of directly and constantly being transmitted through the network. This solution implies a lower bandwidth and energy consumption of the involved nodes. Moreover, the framework is conceived for WSNs that operate on-the-go, hence the Edge Computing paradigm is a perfect fit for the context.

A microserver can operate in all the three layers inasmuch it can also be equipped with sensors. In our testbed this role is covered by the Arduino Uno board, that in the first use case scenario can be seamlessly embedded in a beach bag and can be equipped with a multitude of sensing technology. In this way, the WSN can collect supplementary and more comprehensive data.

Finally, more complex processing takes place at the *Server*, which has the goal of aggregating all data collected by components and microservers, store and process them in order to support the services offered to the user. The server operates only at the Application and Communication layer. As mentioned through Section 3, the server uses the collected data to provide contextualized, personalized services to the end-user.

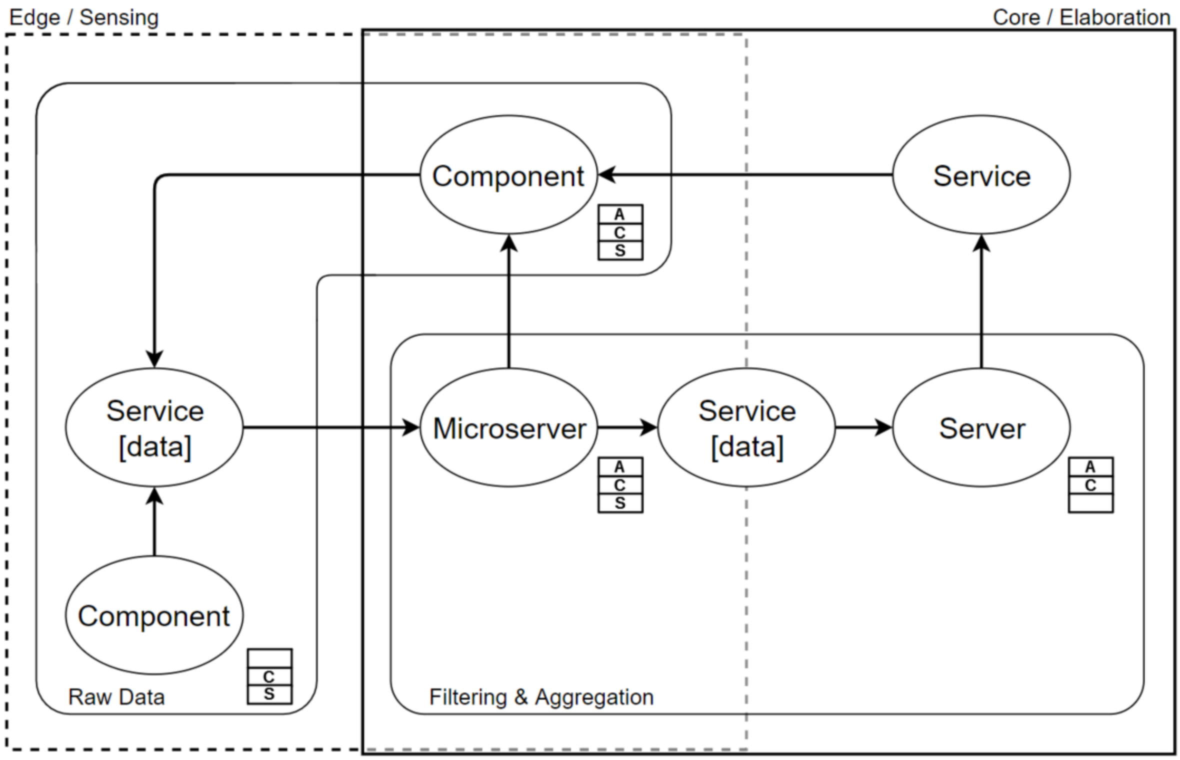


FIGURE 2 WSN framework

4.2 | Interaction Flows

Figure 2 depicts the interaction flows among the different entities of our system. To this end, we have identified three main interaction flows: *components to microserver*, *microserver to server* and *server to components*.

Components to microserver is the flow comprising raw data acquired from the sensors. Components with sensing capabilities perform (a)-periodic (e.g., event-triggered) measurements and the data are sent to the microserver without any processing taking place. Once the microserver receives the data, an optional, lightweight processing phase might take place depending on the type of data under scrutiny. At this stage, the data might be temporarily stored in the microservers memory.

Microserver to server takes place whenever the microserver has to forward data to a server destination. The data at the microserver can undergo a pre-processing stage (e.g., data aggregation), stored locally and successively transmitted. The frequency of this interaction is lower than the previous one so as to limit the energy consumption. Once the data is received by the server, the processing can take place. The result of this processing constitutes yet another service that brings to the third type of flow.

The last interaction, *server to components*, relates to the flow that connects the central server with a user device. In this type of interaction, the server offers not only digested data about, e.g., weather conditions and the environment, but also services, information and warnings. It is noteworthy to point out that the final consumer of this information and services might be both the user and/or another entity who owns (parts of) the infrastructure.

In Figure 2 we make a further distinction between *edge* and *core* nodes. As mentioned before, the edge nodes are the ones located at the logical boundaries of the network such as the components that collect and/or visualize data. The core nodes are positioned in the center of the network and the servers embody an example of these nodes. In our envisioned architectural framework edge nodes are capable of executing lightweight functions on the data while the heavy burden, intensive computations, are offloaded to a server in the core network.

The microserver in fact plays an important role in the data management: once raw data are generated by sensing components, they are sent to the microserver that filters them so as to exclude useless or incorrect information. This operation reduces the amount of information processed at the server. An example of this kind of operations can be the management of the GPS data: in the presented use cases, the microserver collects GPS data and performs a preliminary filtering analysis so as to exclude all the data that present incorrect coordinates (e.g., null coordinates or wrong positioning).

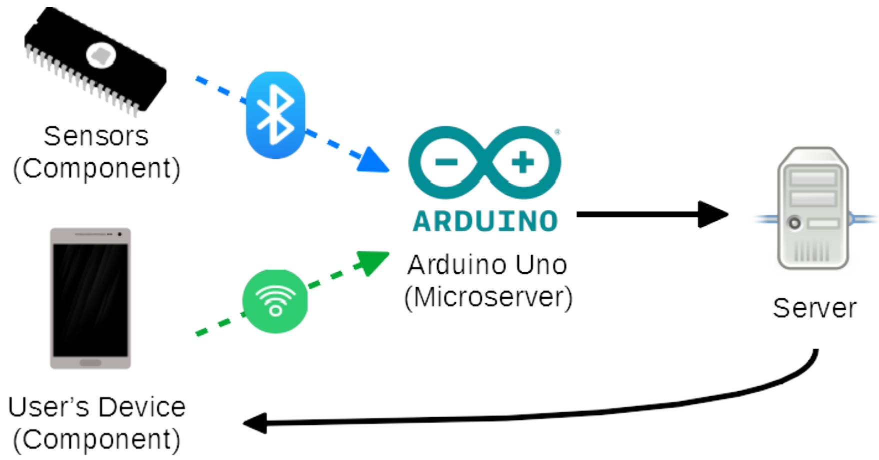


FIGURE 3 General testbed configuration

4.3 | Privacy Issues

The WSN collects user-related data and, for this reason, it is of paramount importance to consider the privacy implications of our proposal. First, it is important to notice that not all the data recorded from the sensors need to be directly attributed to the user. As an example, humidity and temperature data pertain to the environment the user resides in and, given their scope, the system does not necessarily need to associate the sensed data to the user profile and/or its geographical position.

The only sensor that acquires sensible user data that might be used to track and/or infer (in)direct information about the user is the GPS; yet, in this case, the goal is to provide tailored information to end-users. Regardless of the data source and whether it can be directly or indirectly tracked back to the user, privacy in the system can be guaranteed by anonymizing the data at the source and/or adopting organizational and technological policies enforced throughout infrastructure nodes, including but not limited to the centralized units.

From a practical viewpoint, businesses could employ a policy whereby users are attributed temporary anonymous identifiers used to login to the platform, subscribing to the interested featured services. This identifier could be used to store (tag) contextual data. In addition, users might create temporary associations between these identifiers e.g., parents monitoring their children, and this association could be transparent to the end-system.

The use-cases and featured services provided within a scenario can differ and the user should be able to subscribe to a service depending on the service agreement. At the same time, the envisaged system should provide the user with the possibility to choose the scope of the generated data, e.g., time validity and the possibility to opt out from the service.

4.4 | A Working Example

While preserving the general aspect of this study and without loss of generality, in the following, we instantiate the WSN to the beach resort scenario showcasing an instance of the depicted framework. We assume that the targeted beach resort offers a set of gadgets to its clients such as hats or bracelets, equipped with sensors capable of acquiring the users' position, e.g., through GPS technology, data about temperature, UV ray and humidity, etc. All the data are collected and sent to the server of the seaside resort. The server, after having processed the data, advertises recommendations and promotions based, for instance, on the weather conditions.

In Figure 3, we depict the implemented WSN using Arduino boards embodying a set of sensors specifically selected for weather monitoring. We employed the TCP protocol for data transfer and the code run in the microserver to collect data from sensors is implemented using the C++ language.

In addition to the sensing layer, we provision a centralized server, implemented with a *LAMP (e.g., Linux, Apache, MySQL, PHP)* architecture, used to gather and elaborate the data, offering contextualized, personalized services to the customers through, e.g., a smartphone app. It is noteworthy to point out that the system allows different service models, that is either push or pull service models.

To prepare the testbed, we evaluated various elements on the basis of their characteristics and suitability for the selected use case. This process led us to choose the boards and sensors presented in Appendix A among a wide range of possible choices.

The main board we employed is an Arduino Uno Rev. 3, equipped with the Bluetooth HC-05 and the Wi-Fi ESP8266 modules. This board acts as the microserver and has the task of handling the communication with the other boards equipped with sensors, which are an Arduino LilyPad Main

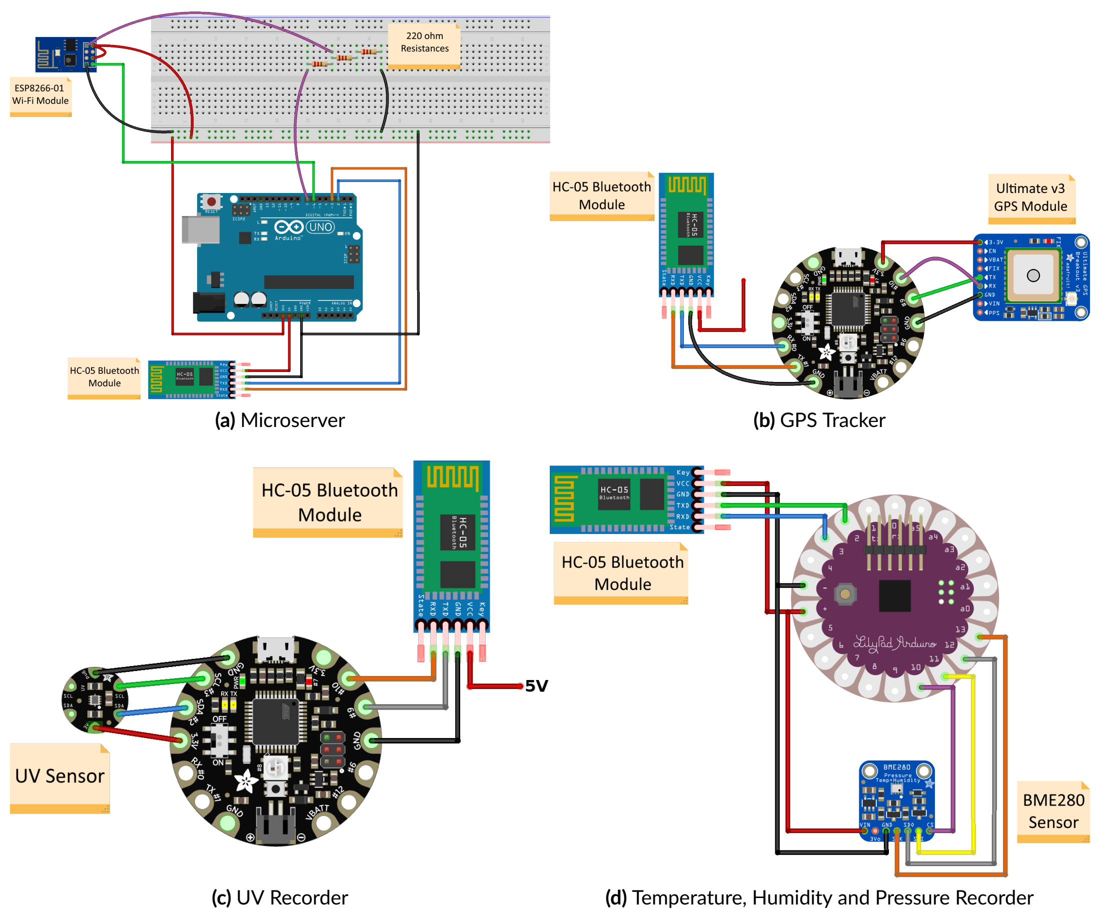
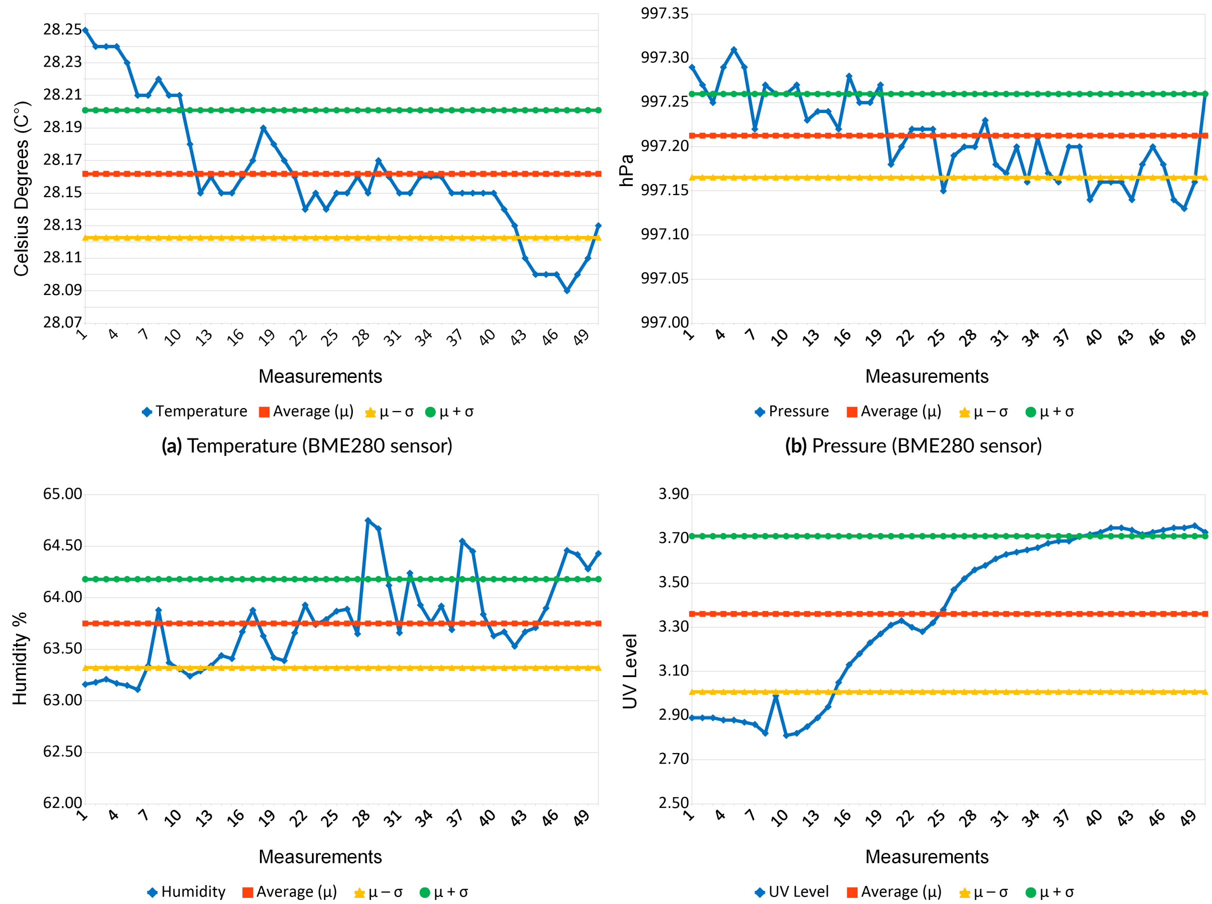


FIGURE 4 Detailed testbed configuration

Board and Adafruit Flora, representing the components of the network specifically designed as wearable devices that can be applied to clothes. They both have sensors attached used to collect data from the environment and a Bluetooth HC-05 module for communication. The sensors that have been selected are the Si1145 sensor for the UV rays, BME 280 sensor for pressure, temperature and humidity, and the Flora Wearable Ultimate GPS module for the location tracking. A detailed visual description of the configuration of the testbed is shown in Figure 4.

5 | TEST PHASE AND RESULTS

We performed a set of field trials to assess the feasibility and reliability of the system. In specific, the goal is to asses the system capabilities in terms of accuracy of the sensed data and energy expenditure of the individual components. To this end, we collected results for the sensors used for the testbed: (i) BME280 for temperature, humidity and pressure, (ii) Si1145 for the UV level and (iii) Ultimate V3 GPS Module for the GPS tracking. Initial testing showed that some sensors are not fully reliable; we hence implemented customized algorithms to filter and pre-process sampled data before transmitting them to a destination. We point out that the data have been recorded and collected in an open environment. This is done in order to emulate realistic conditions pertaining to our case studies such as, for instance, a user in a park, moving from an open space to another location where the measured phenomena is partially or entirely obstructed (e.g., under a tree).



(a) Temperature (BME280 sensor)

(b) Pressure (BME280 sensor)

(c) Humidity (BME280 sensor)

(d) UV Level (Si1145 sensor)

FIGURE 5 Measurements performed by the BME280 and Si1145 sensors

5.1 | Environmental Measurements

We recorded various environmental parameters such as temperature, humidity, pressure and UV level in order to consider a typical working day of the system. For each environmental parameter, we performed 50 measurements with a 10 s time lapse. The obtained results are presented in Figure 5: each graph shows the collected data for each parameter (i.e., temperature in Figure 5a, atmospheric pressure in Figure 5b, humidity in Figure 5c and UV level in Figure 5d), the mean values and the standard deviation to highlight how much data vary during the measurements.

Figures 5a, 5b and 5c show the data collected by the BME280 sensor. The first and the second graphs show that sampled data for the temperature and pressure parameters are homogeneous, presenting few variations between the data points. Figure 5c instead shows wider oscillations with respect to the other two parameters; nevertheless, they remain limited and lower than one percentage point.

Figure 5d reports the UV values collected by the Si1145 sensor, recording the light intensity both in the visible and infrared spectrum. On the basis of these data, it computes an estimation of the UV level. The performance of Si1145 is generally good, with a smooth increment of the UV level and a moderate standard deviation.

5.2 | GPS Measurements

The GPS tracking data has been collected using the Ultimate v3 GPS Module and the tests have been conducted at different locations so as to resemble a real scenario. In particular, we recorded the GPS position far from obstacles (see Table B6 in Appendix B) and in proximity of sources of interference such as trees and buildings (see Table B7 in Appendix B) . Each test is composed by 100 position measurements; however, for lack of space, only a subset of them has been reported in Table 1.

Localization error of the GPS module				
Date	**Time**	**Real Position**	**Detected Position**	**Error (m)**
04/06/17	18:25:40	(45.892815,12.082536)	(45.89283,12.082583)	5.11
04/06/17	18:25:50	(45.89283,10.415869)	(45.89283,12.082583)	128983.9

TABLE 1 Error between the real position of the GPS module and the detected one

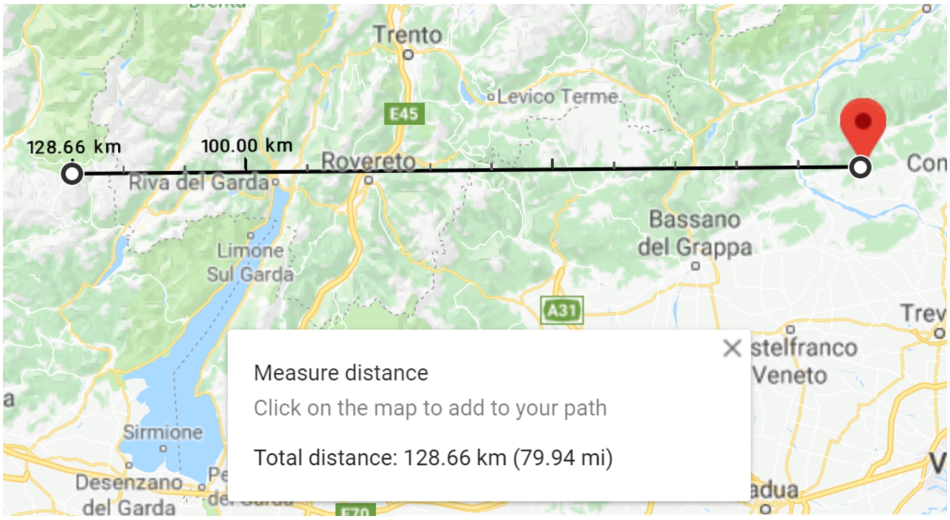


FIGURE 6 Error of the GPS module, map representation

The wearable sensors are usually hardware components built to minimize not only the size of the hardware itself, but also the energy consumption and cost. This can lead to a lower precision of the devices, hence some pre-processing of the sensed data is often necessary. During our tests, the GPS module introduced some errors in the measurements; therefore, it is the best candidate to describe the filtering elaboration performed by the microserver of the WSN.

In our testbed, the pre-processing, filtering step of the sensed data is performed by the Arduino Uno board, i.e., the microserver. An example of a typical situation of incorrect data that we actually encountered during our experiments can be observed in Table 1 and it is better visualized in Figure 6: we can see two measurements made within a 10 s time interval and that represent a movement of more than 120 km. This is clearly impossible, since this movement would require a too high speed of the users, thus evidencing an error introduced by the GPS module. The microserver is then in charge to apply some filters to clean the data. The filters utilized in our measurements check the validity of the speed required by the movement.

These computations precede more complex operations executed by the centralized servers. This simple, preliminary elaboration could also allow to lower the computational load of the centralized servers, avoiding the transmission of useless data.

5.3 | Energy Consumption Measurements

Communication, sensing and processing are all contributing to battery consumption. Once deployed, the sensing platform should have sufficient autonomy under reasonable duty cycling regimes. To this end, we asses the energy expenditure of each component through a measurement instrument, the Power Monitor produced by Monsoon Solutions Inc., which allows to power an electronic device and, at the same time, measure many fundamental parameters such as tension, current and absorbed power. For each of these parameters the minimum, maximum and average values are specified for the whole duration of the measurement. Using the power monitor it is then possible to understand if a certain type of battery is sufficient for the monitored device or if a component is not feasible because it requires a bulky battery.

We measure the power consumption of all the various components of the wearable testbed and define the following groups of devices as evidenced in Figure 4:

- **Arduino** + HC-05 BT Module + ESP8266 Wi-Fi Module

Energy consumption of the four groups of devices				
	Arduino + BT + Wi-Fi	**LilyPad + BT + °C/%/hPa**	**Flora + BT + UV**	**Flora + BT + GPS**
Energy Consumption [mAh]	63.6	4.6	3.9	8.9
Average Power [mW]	944.82	51.38	57.98	132.30
Average Voltage [V]	3.71	3,71	3,71	3,71
Battery Capacity [mAh]	2200	100	350	610
Estimated Battery Life [h]	8.65	7.23	22.42	17,12

TABLE 2 Energy consumption of the four boards equipped with the related sensors and the communication modules

- **LilyPad** + HC-05 BT Module + BME280 temperature, humidity, atmospheric pressure Module
- **Flora** + HC-05 BT Module + Si1145 UV Module
- **Flora** + HC-05 BT Module + Ultimate Wearable GPS Module

For every group of devices we perform two measurements which lasted 900 s each.

Table 2 presents the power consumption of the four groups of devices adopted in our testbed: all the components show a good performance and the chosen battery equipment is adequate, amounting to acceptable operational times under reasonable duty cycling regimes. In specific, we adopted small batteries for the sensing components (LilyPad, Flora) to contain the fabric size as they have to be attached to clothes and wearable accessories. The microserver instead (i.e., Arduino Uno) is equipped with a more powerful battery in order to allow also for more onerous computational operations. Moreover, the idea is to embed the microserver into bigger accessories, like bags or backpacks, that can easily contain it without impeding their usability.

The measurements show that the estimated battery life of the four hardware bundles varies from circa 7 hours to 22. These profiles are more than sufficient for a daily usage in both the envisioned scenarios. Also, it is noteworthy to point out, that no particular optimization study was involved and there is room for additional improvement in terms of battery lifetime and associated duty cycling regimes. As an example, the sole component that can represent a problem is the LilyPad board with a lifetime of only 7 hours. However, we must note here that it is equipped with a very small battery (100 mAh) which can be easily replaced with a battery of 350 mAh capacity with a limited increase in size.

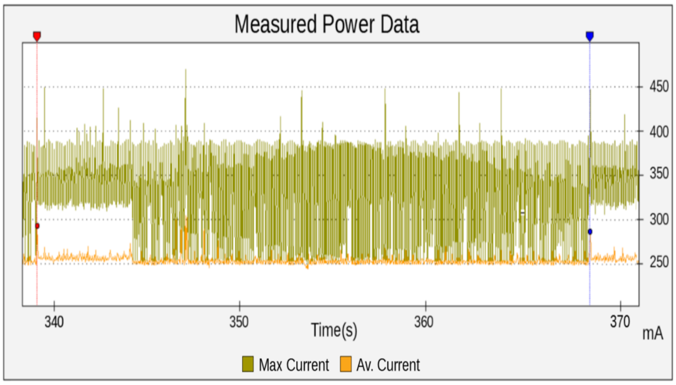
Figures 7a, 7b, 7c and 7d show a detailed report of the energy consumption during the measurements. Figure 7a reports energy consumption of the microserver; in particular, the orange curve represents the average consumption and the green curve represents the maximum consumption. We can note that there are peaks of absorbed energy when the Bluetooth module receives the data from the sensors (red marker on the left) and when the Wi-Fi module sends data to the centralized servers (blue marker on the right). This means that the two modules are responsible for the increased energy consumption, therefore a good balancing of the sleep-mode and working phases is crucial for a good battery management.

Figure 7c reports energy consumption of the sensing component which collects data about atmospheric pressure, temperature and humidity and sends this data to the microserver through Bluetooth. We can observe that the energy consumption is initially low, then it increases for a certain time interval and then returns to the initial level. This increment is due to the use of the Bluetooth module and to the transmission of sampled data from the sensor. In this interval of time, we can notice three different working regime areas: the red ones located on the left and right sides (denoted in the figure by the numbers 1 and 3) contain some small peaks, and the green one (denoted in the figure by the number 2) with a more linear trend. These three zones correspond to three different moments during the data transmission. In specific, zone 1 corresponds to the reception by the Bluetooth module of the ID code sent by the microserver that triggers the detection by the sensor, zone 2 corresponds to the effective reading of the values detected by the sensor and, finally, zone 3 corresponds to the shipping of the values just read to the Arduino Uno microserver. It is noteworthy to point out that the continuous and regular pulsations visible on the measurement graph are caused by the LED of the Bluetooth module and of the LilyPad board itself.

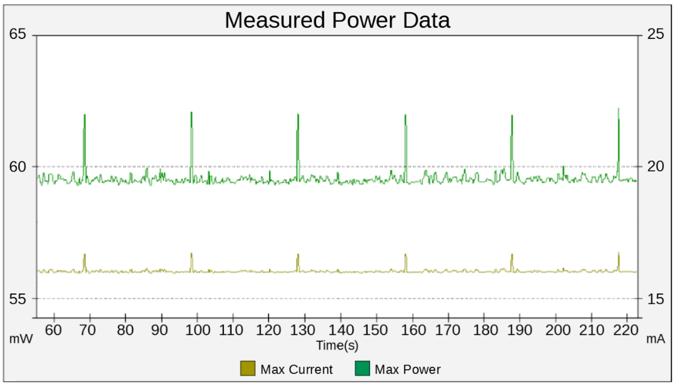
As shown in Figure 7b, the energy consumption of the Flora Board equipped with the Bluetooth and UV modules is slightly different with respect to the others. It can be observed that there are some clear peaks in correspondence to the data acquisition from the sensor and the subsequent transmission of the latter. The Si1145 is a sensor that detects only one parameter and this leads to a shorter time interval between reading and transmission. Therefore, there are no particularly wide plateau like in the previous case, thus the specific shape of the observed graph.

The graph in Figure 7d shows the trend of the Flora board equipped with the Bluetooth and GPS modules. It can be seen that during the acquisition of the GPS coordinates, the communication between the GPS module and the GPS satellites greatly increases the energy consumption. The update frequency of the Ultimate Wearable GPS Module is set to a 15 s interval and it can be perfectly observed in the zones 1 and 2 of the figure.

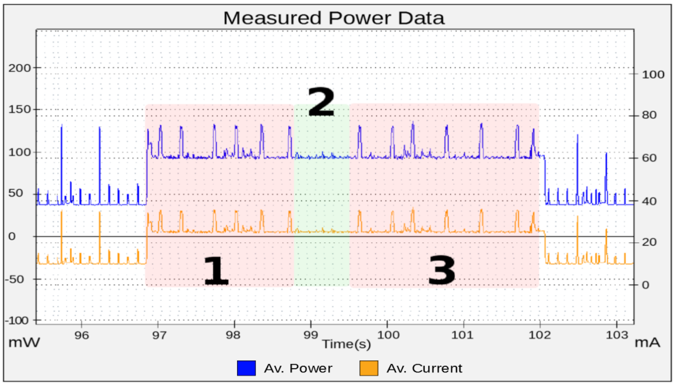
Apart from some small differences, all the nodes of the network present a similar behavior in their energy profiles. This confirms that the more energy-demanding operations are related to the transmission of messages between the nodes and the update of the GPS position. It is then crucial



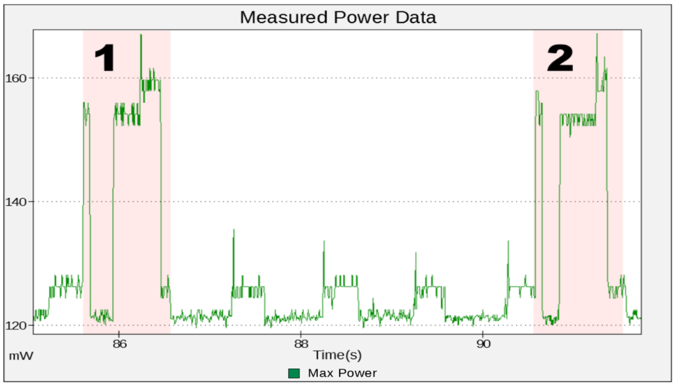
(a) Details of the energy consumption of the group of devices composed by the Arduino Uno board, the HC-05 Bluetooth Module and the ESP8266 module



(b) Details of the energy consumption of the group of devices composed by the Flora board, the HC-05 Bluetooth Module and the Si1145 UV sensor



(c) Details of the energy consumption of the group of devices composed by the LilyPad board, the HC-05 Bluetooth Module and the BME280 sensor (pressure, temperature, humidity)



(d) Details of the energy consumption of the group of devices composed by the LilyPad board, the HC-05 Bluetooth Module and the Ultimate Wearable GPS Module

FIGURE 7 Measurements performed by the BME280 and Si1145 sensors

to perfectly balance the communications in the WSN because its efficiency is closely related to them. Nevertheless, the results obtained by our testbed are quite promising as already shown in Table 2.

6 | CONCLUSION

This work presents a framework able to model a WSN also considering network architectures with heterogeneous nodes. The WSN is conceived to collect information about the environment and to provide services to the users; to make this possible we designed the framework simple and flexible, so it can be adapted to different use cases but with similar base requirements.

We choose to add microservers to our architecture because this kind of approach allows to deploy small services that can scale independently of each other [32,33]. Microservers overcome the shortcomings of monolithic architectures thus facilitating scalability [34]. However, this kind of networks also presents some notorious problems such as the calibration of sensors and battery life. We analyze and try to mitigate these problems adopting the concept of Edge Computing that suggests to provide computing services for customers in the space between network central servers and end-users.

To evaluate the framework, we created a testbed on the basis of one of the two discussed use cases and subsequently performed a series of experiments. The first set of experiments has the goal to verify whether the testbed can detect environment parameters and generate meaningful results, cleaning the raw data produced by the sensors from errors and inaccurate measurements. Instead, the second set is focused on the energy consumption of the WSN during an hypothetical day of usage to analyze whether the framework is effectively applicable on a network of wearable devices.

	Arduino Uno	Arduino LilyPad	Adafruit Flora
Processor	ATmega328P	ATmega328P	ATmega32u4
Labour Input Voltage [V]	5 / 7 - 12	2.7 - 5.5	3.3 / 3.6 - 6
CPU Speed [Mhz]	16	8	8
EEPROM [kB]	1	512	n.d.
SRAM [kB]	2	1	n.d.
Flash Memory [kB]	32	16	n.d.
USB	type B	–	micro
Built-in connectivity	–	–	–
Size [mm]	69 x 54	50 (ϕ) x 8	45 (ϕ) x 7
Energy Consumpt. @ 5V [mA]	$\sim$47 in exec. $\sim$35 in standby	N/A	N/A

TABLE A1 Main technical specifications on the adopted microcontroller boards

Bluetooth Module HC-05	
Transmissive Power [dBm]	$\leq$4, class 2
Bit rate [Mbps]	2.1/0.16 (Asynchronous) 1/1 (Synchronous)
Labour Input Volt. [V]	5 / 3.3
Working Temp. [°C]	-20 $\sim$ 55
Size [mm]	26.9 x 13 x 2.2
Energy Consumpt. [mA]	8 in exec. <30 in pairing

TABLE A2 Main technical characteristics of the Bluetooth HC-05 module

The results presented in this paper show the capability of the WSN to produce meaningful information about, for example, the GPS position in both an open environment and in presence of obstacles by recognizing the incorrect data introduced by the sensor. After that, the testbed built on the basis of the proposed framework presents an adequate energy consumption of the different nodes of the network, showing that a daily use is indeed possible.

The performances obtained from the testbed are quite promising and show that the presented framework can be a good choice for a network of wearable sensors and it can be applied to a wide variety of scenarios. Moreover, the testbed described in this study is a prototype that can be further optimized from both a software and hardware point of view. It would then be possible to obtain even better results in a perfected commercial version.

ACKNOWLEDGEMENTS

The authors would like to thank Filippo Damuzzo for his technical contribution to the development of the system.

How to cite this article: A. Bujari, O. Gaggi, C. E. Palazzi and G. Quadrio (2019), Wearable Sensor Networks: A Measurement Study, *CCPE, 2019;00:1–6*.

APPENDIX

A TECHNICAL SPECIFICATIONS OF CHOSEN BOARDS AND SENSORS

To prepare the testbed, we evaluated various components on the basis of their characteristics and suitability for the selected use case. This process led us to choose the boards and sensors presented in Tables A1, A2 A3, A4 and A5 among a wide range of possible choices.

ESP8266 mod. 01	
Supported Standards	802.11 b/g/n, Wi-Fi Direct
Security	WPA/WPA2
Labour Input Volt. [V]	3.3/3 - 3.6
CPU	low power 32-bit @ 80 Mhz
ROM-bootloader [KB]	64
RAM [KB]	64 (instructions) 96 (data)
Flash Memory [MB]	1
Serial Connection	SPI, I2C, UART
Size [nn]	25 x 38 x 5
Energy Consumpt. [mA]	$\sim$60 - 215 in exec. $\sim$0.0009 in standby

TABLE A3 Main technical characteristics of the Wi-Fi ESP8266-01 module

Si1145 Sensor for Flora	
Labour Input Voltage [V]	1.7 - 3.6/3.3
Working Temp. [$^\circ$C]	-40 - 85
IR Spectrum Range [nm]	550 - 1000 (centered in 800)
Visible Light Spectrum Range [nm]	400 - 800 (centered in 530)
Serial Connection	I2C
Size [mm]	14 (ϕ)
Energy Consumption [mA]	$\sim$0.009 in exec. $<$0.0005 in standby

TABLE A4 Main technical characteristics of the Si1145 sensor for the UV detection by Adafruit. *Flora Version*

BME 280 Sensor	
Labour Input Voltage [V]	3 - 5
Pressure Detection Range [hPa]	300 - 1100
Pressure Accuracy [hPa]	$\pm$ 1
Temperature Detection Range [$^\circ$C]	-40 - 85
Temperature Accuracy [$^\circ$C]	$\pm$ 1
Humidity Detection Range [%]	0 - 100
Humidity Accuracy [%]	$\pm$ 3
Serial Connection	I2C, SPI
Size [mm]	18 x 19 x 2
Energy Consumption [mA]	0.0018 @ 1 Hz humid./temp. readings 0.0028 @ 1 Hz press./temp. readings 0.0036 @ 1 Hz humid./press./temp. readings 0.0001 in standby

TABLE A5 Main technical characteristics of the BME280 sensor by Adafruit.

B DATA RECORDED FROM GPS

Tables B6 and B7 show an instance of filtered data collected by the GPS sensor, respectively, in absence of obstacles and in presence of trees. As we can see, the average error is 1.93 m in an open area and grows to 6.66 m in presence of trees. In both cases, the values are acceptable for our use cases, because a distance of at most 7 m is acceptable both for kids' surveillance and as area of research for survivors.

Detected positions of the GPS module			
Time	**GPS Position**	**Reference Position**	**Error (m)**
09:08:28	(45.895530, 12.082282)	(45.895516, 12.082298)	2.16
09:08:33	(45.895530, 12.082278)	(45.895516, 12.082298)	2.35
09:08:37	(45.895523, 12.082274)	(45.895516, 12.082298)	2.10
09:08:43	(45.895530, 12.082270)	(45.895516, 12.082298)	2.79
09:08:53	(45.895542, 12.082270)	(45.895516, 12.082298)	3.70
09:09:08	(45.895530, 12.082272)	(45.895516, 12.082298)	2.67
09:09:11	(45.895530, 12.082267)	(45.895516, 12.082298)	2.97
09:09:23	(45.895530, 12.082265)	(45.895516, 12.082298)	3.09
09:09:28	(45.895523, 12.082270)	(45.895516, 12.082298)	2.37
09:09:32	(45.895523, 12.082272)	(45.895516, 12.082298)	2.23
09:09:36	(45.895523, 12.082278)	(45.895516, 12.082298)	1.83
09:09:49	(45.895515, 12.082294)	(45.895516, 12.082298)	0.37
09:09:52	(45.895507, 12.082296)	(45.895516, 12.082298)	0.88
09:10:04	(45.895507, 12.082304)	(45.895516, 12.082298)	0.96
09:10:09	(45.895507, 12.082315)	(45.895516, 12.082298)	1.52
09:10:14	(45.895507, 12.082312)	(45.895516, 12.082298)	1.34
09:10:19	(45.895507, 12.082308)	(45.895516, 12.082298)	1.12
09:10:29	(45.895515, 12.082312)	(45.895516, 12.082298)	1.03
09:10:48	(45.895500, 12.082308)	(45.895516, 12.082298)	1.85
09:10:53	(45.895507, 12.082312)	(45.895516, 12.082298)	1.34
	Average error		**1.93**
	Standard Deviation		**0.843**

TABLE B6 Detected GPS positions in an open environment, without obstacles

Detected positions of the GPS module			
Time	**GPS Position**	**Reference Position**	**Error (m)**
08:46:52	(45.895679, 12.082066)	(45.895626, 12.082060)	5.96
08:46:57	(45.895687, 12.082068)	(45.895626, 12.082060)	6.81
08:47:02	(45.895687, 12.082077)	(45.895626, 12.082060)	6.90
08:47:07	(45.895687, 12.082093)	(45.895626, 12.082060)	7.24
08:47:17	(45.895679, 12.082132)	(45.895626, 12.082060)	8.12
08:47:32	(45.895679, 12.082121)	(45.895626, 12.082060)	7.59
08:47:37	(45.895679, 12.082121)	(45.895626, 12.082060)	7.59
08:47:53	(45.895687, 12.082111)	(45.895626, 12.082060)	7.83
08:47:53	(45.895687, 12.082111)	(45.895626, 12.082060)	7.83
08:48:06	(45.895679, 12.082095)	(45.895626, 12.082060)	6.51
08:48:09	(45.895679, 12.082093)	(45.895626, 12.082060)	6.45
08:48:17	(45.895687, 12.082089)	(45.895626, 12.082060)	7.14
08:48:21	(45.895679, 12.082089)	(45.895626, 12.082060)	6.34
08:48:27	(45.895679, 12.082010)	(45.895626, 12.082060)	7.11
08:48:38	(45.895671, 12.082075)	(45.895626, 12.082060)	5.21
08:48:43	(45.895671, 12.082073)	(45.895626, 12.082060)	5.18
08:48:48	(45.895671, 12.082073)	(45.895626, 12.082060)	5.18
08:48:53	(45.895679, 12.082077)	(45.895626, 12.082060)	6.07
08:49:01	(45.895679, 12.082077)	(45.895626, 12.082060)	6.07
08:49:07	(45.895679, 12.082077)	(45.895626, 12.082060)	6.07
	Average error in m		**6.66**
	Standard Deviation		**0.884**

TABLE B7 Detected GPS positions in an environment with obstacles (trees)

References

1. Thorp EO. The Invention of the First Wearable Computer. In: *Proc. of the International Symposium on Wearable Computers*; 1998; Washington, DC, USA: 4–8.

2. Montori F, Bedogni L, Bononi L. A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring. *IEEE Internet of Things Journal* 2018; 5(2): 592-605.

3. Razzaque MA, Hira MT, Dira M. QoS in Body Area Networks: A Survey. *ACM Transactions on Sensor Networks* 2017; 13(3): 25:1–25:46. doi: 10.1145/3085580

4. Chouikhi S, Korbi IE, Ghamri-Doudane Y, Saidane LA. Recovery from simultaneous failures in a large scale wireless sensor network. *Ad Hoc Networks* 2017; 67: 68 - 76.

5. Bujari A, Ciman M, Gaggi O, Marfia G, Palazzi CE. PathS: Enhancing Geographical Maps with Environmental Sensed Data. In: *Proc. of the Workshop on Pervasive Wireless Healthcare*; 2015; Hangzhou, China: 13–16.

6. Alvear O, Calafate CT, Zema NR, et al. A Discretized Approach to Air Pollution Monitoring Using UAV-based Sensing. *Mobile Networks and Applications* 2018; 23(6): 1693–1702.

7. Boudway I. Is Chris Dancy the Most Quantified Self in America?. 2014.

8. Ang LM, Seng KP, Zungeru AM, Ijemaru GK. Big Sensor Data Systems for Smart Cities. *IEEE Internet Things Journal* 2017; 4(5): 1259–1271.

9. Scataglini S, Andreoni G, Gallant J. A Review of Smart Clothing in Military. In: *Proc. of the ACM Workshop on Wearable Systems and Applications*; 2015: 53–54.

10. Perego P, Moltani A, Andreoni G. Sport Monitoring with Smart Wearable System. In: *Proc. of Int. Conf. WearableMicro Nano Technol. Pers. Heal.*; 2012: 222–224.

11. Patel S, Park H, Bonato P, Chan L, Rodgers M. A Review of Wearable Sensors and Systems with Application in Rehabilitation. *Journal of NeuroEngineering and Rehabilitation* 2012; 9(1): 21.

12. Matsuoka S, Ogawa H, Maki H, Yonezawa Y, Caldwell M. A New Safety Support System for Wandering Elderly Persons. In: *Proc. of the IEEE Engineering in Medicine and Biology Society*; 2011: 5232–5235.

13. Kolasinska A, Quadrio G, Gaggi O, Palazzi CE. Technology and Aging: Users' Preferences in Wearable Sensor Networks. In: *Proc. of the 4th EAI International Conference on Smart Objects and Technologies for Social Good*; 2018: 77–81.

14. Campbell AT, Eisenman S, Lane N, Miluzzo E, Peterson R. People-centric Urban Sensing. In: *Proc. of the ACM International Workshop on Wireless Internet*; 2006; New York, NY, USA.

15. Roccetti M, Marfia G, Palazzi CE. Entertainment beyond divertissment: using computer games for city road accessibility. *ACM Computers in Entertainment* 2011; 9(2).

16. Prandi C, Mirri S, Ferretti S, Salomoni P. On the need of trustworthy sensing and crowdsourcing for urban accessibility in smart city. *ACM Transactions on Internet Technology* 2017; 18(1).

17. DeMichele R, Furini M, Montangero M. Crowd-sensing Images to Understand Citizens' Emotions, Issues and Interests. In: *Proc. of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*; 2019.

18. Pozza M, Rao A, Bujari A, Flinck H, Palazzi CE, Tarkoma S. A refactoring approach for optimizing mobile networks. In: *Proc. of the IEEE International Conference on Communications (ICC)*; 2017.

19. Delnevo G, Di Lena P, Mirri S, Prandi C, Salomoni P. On combining Big Data and machine learning to support eco-driving behaviours. *Journal of Big Data* 2019; 6(64): 1–15.

20. Roccetti M, Delnevo G, Casini L, Cappiello G. Is bigger always better? A controversial journey to the center of machine learning design, with uses and misuses of big data for predicting water meter failures. *Journal of Big Data* 2019; 6(70): 1–23.

21. Akpakwu GA, Silva BJ, Hancke GP, Abu-Mahfouz AM. A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. *IEEE Access* 2018; 6: 3619–3647.

22. Bujari A, Furini M, Mandreoli F, Martoglia R, Montangero M, Ronzani D. Standards, security and business models: key challenges for the IoT scenario. *Mobile Networks and Applications* 2018; 23(1): 147–154.

23. Jutila M, Strömmer E, Ervasti M, Hillukkala M, Karhula P, Laitakari J. Safety services for children: a wearable sensor vest with wireless charging. *Personal and Ubiquitous Computing* 2015; 19(5): 915–927.

24. Umek A, Tomažič S, Kos A. Wearable training system with real-time biofeedback and gesture user interface. *Personal and Ubiquitous Computing* 2015; 19(7): 989–998.

25. Elena Agapie *et. al* . Seeing Our Signals: Combining Location Traces and Web-based Models for Personal Discovery. In: *Proc. of the ACM Workshop on Mobile Computing Systems and Applications*; 2008; New York, NY, USA: 6–10.

26. Tse R, Pau G. Enabling street-level pollution and exposure measures: a human-centric approach. In: *Proc. of the 6th ACM International Workshop on Pervasive Wireless Healthcare (MobiHealth'16)*; 2016.

27. Gaonkar S, Li J, Choudhury RR, Cox L, Schmidt A. Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation. In: *Proc. of the ACM International Conference on Mobile Systems, Applications, and Services*; 2008; New York, NY, USA: 174–186.

28. Aram S, Troiano A, Pasero E. Environment sensing using smartphone. In: *Proc. of the IEEE Sensors Applications Symposium Proceedings*; 2012: 1–4.

29. Das T, Mohan P, Padmanabhan VN, Ramjee R, Sharma A. PRISM: Platform for Remote Sensing Using Smartphones. In: *Proc. of the International Conference on Mobile Systems, Applications, and Services*; 2010: 63–76.

30. Belli L, Cirani S, Gorrieri A, Picone M. A Novel Smart Object-Driven UI Generation Approach for Mobile Devices in the Internet of Things. In: *Proc. of the ACM Workshop on Experiences with the Design and Implementation of Smart Objects*; 2015; New York, NY, USA: 1–6.

31. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things Journal* 2017; 4(5): 1125–1142.

32. Dragoni N, Giallorenzo S, Lafuente AL, et al. *Microservices: Yesterday, Today, and Tomorrow*: 195–216; Springer International Publishing . 2017.

33. Melis A, Mirri S, Prandi C, Marco Prandini FC. Integrating Personalized and Accessible Itineraries in MaaS Ecosystems Through Microservices. *Springer Mobile Networks and Applications* 2018; 23(1): 167–176.

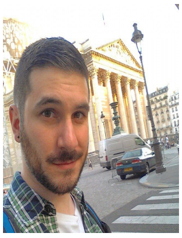
34. Hasselbring W, Steinacker G. Microservice Architectures for Scalability, Agility and Reliability in E-Commerce. In: *Proc. of IEEE International Conference on Software Architecture Workshops (ICSAW)*; 2017: 243-246.

AUTHOR BIOGRAPHY

ARMIR BUJARI (abujari@math.unipd.it) is an Assistant Professor in Computer Science at the University of Padua. He received his Ph.D. degree in Computer Science at the University of Bologna, Italy, in 2014. His research interests are primarily focused on the design and analysis of communication protocols for wired/wireless networks, network architectures, and delay-tolerant communication, with an emphasis on distributed sensing, mobile applications and multimedia entertainment.

OMBRETTA GAGGI (gaggi@math.unipd.it) is an Assistant Professor in Computer Science at the Department of Mathematics of the University of Padua. She received the M.S. degree cum laude in Computer Science in December 1998 from University of Venice. In 2003, she received her Ph.D. in Computer Science from the Consortium of Universities of Bologna, Padova and Venice. Her researcher interests include accessibility and web technologies, cross-platform mobile development and user interface design. She is serving as reviewer and program committee member in several international conferences and journal.

CLAUDIO E. PALAZZI (cpalazzi@math.unipd.it) is an Associate Professor in Computer Science at the Department of Mathematics of the University of Padua. He received his M.S. degree in Computer Science from UCLA in 2005, his Ph.D. degree in Computer Science from UniBO in 2006, and his Ph.D. degree in Computer Science from UCLA in 2007. His research interests are primarily focused on the design and analysis of communication protocols for wired/wireless networks, Internet architectures, and mobile users, with an emphasis on distributed sensing, mobile applications and multimedia entertainment. He is active in various technical program committees of the most prominent international conferences and is author of more than 150 papers, published in international conference proceedings, books, and journals.



GIACOMO QUADRIO (gquadrio@math.unipd.it) is a Ph.D. student in Computer Science at the Department of Mathematics of the University of Padua. He received his M.S. degree in Computer Science from the University of Padua in 2016. His research interests are primarily focused on the design and analysis of communication protocols for wired/wireless networks, network architectures and mobile users, with an emphasis on mobile applications and multimedia entertainment.