

Research Article

AAL Middleware Infrastructure for Green Bed Activity Monitoring

Filippo Palumbo,^{1,2} Paolo Barsocchi,¹ Francesco Furfari,¹ and Erina Ferro¹

¹ *ISTI-CNR, Pisa Research Area, Via G. Moruzzi 1, 56124 Pisa, Italy*

² *Computer Science Department, University of Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy*

Correspondence should be addressed to Paolo Barsocchi; paolo.barsocchi@isti.cnr.it

Received 8 February 2013; Revised 19 June 2013; Accepted 20 June 2013

Academic Editor: Ignacio Matias

Copyright © 2013 Filippo Palumbo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper describes a service-oriented middleware platform for ambient assisted living and its use in two different bed activity services: bedsore prevention and sleeping monitoring. A detailed description of the middleware platform, its elements and interfaces, as well as a service that is able to classify some typical user's positions in the bed is presented. Wireless sensor networks are supposed to be widely deployed in indoor settings and on people's bodies in tomorrow's pervasive computing environments. The key idea of this work is to leverage their presence by collecting the received signal strength measured among fixed general-purpose wireless sensor devices, deployed in the environment, and wearable ones. The RSS measurements are used to classify a set of user's positions in the bed, monitoring the activities of the user, and thus supporting the bedsores and the sleep monitoring issues. Moreover, the proposed services are able to decrease the energy consumption by exploiting the context information coming from the proposed middleware.

1. Introduction

The last few years have seen research development in the field of ambient assisted living (AAL), which can be defined as concepts, products, and services supporting a healthy and independent life of elderly citizens with intelligent systems that assist them in carrying out their daily activities. AAL encompasses a wide range of applications ranging from tele-monitoring of vital parameters for patients with chronic diseases to scenarios involving home automation and domotics, the recognition of adverse events such as a fall causing a fracture, or specific assistance systems for people with hearing or vision deficits. These researches were focused on network infrastructures, distributed software architectures as well as context information models to support pervasive computing applications in smart environments.

The AAL environments leveraging smart devices have the ability to support user's daily life activities through efficient context evaluation systems that support activities for different users' requirements. At the same time, application adaptation

for these activities is also required in response to changes from the environment. In this regard, the interconnections among components sharing the same context are also relevant. A crucial role in this scenario is played by the middleware infrastructure as it provides the central connection point that is shared by all the components according to the needed information exchanges. A middleware infrastructure provides a set of basic services for the development of an AAL and vital signs monitoring applications.

Activity recognition is an important issue for healthcare since sufficient information about patients is vital for an effective care. Monitoring the activities of patients enables hospital staffs to provide specialized care. For example, in a pervasive hospital, a nurse can use a mobile activity monitor to provide immediate care for patients in need of assistance or in risky situations [1]. Also in home environment, due to decline in both physical and mental abilities, some elderly are often unable to make the desirable bodily movements in the bed and repositioning that are critical for blood circulation and relieving of prolonged pressure over the body. For these

reasons, continuous observation of the patients, through a *bed position detection* service, is necessary in order to prevent the above-mentioned adverse effects. Moreover, monitoring the body movements during the sleep is important to recognize sleeping disorders for diagnosis and prompt treatment of disease. A bed position detection service can also provide detailed sleeping profiles that depict periods of restlessness and interruptions such as bed exits and entries due to visiting the bathroom. This information helps find trends that correlate to certain diseases. Moreover, it enables monitoring effectiveness of treatments to sleep-related diseases. In order to monitor context and behaviour of subjects in the bed, the authors in [2] propose the use of pressure sensing system. In fact, the pressure evidences can assist in determining the elderly persons position. However, this solution is based on specific hardware and does not exploit the pervasive smart environment around the elderly person. Instead, in this work we verified that the presence of a generic and not specific wireless devices (such as light, switch, and mains power outlet sensors) can be exploited to infer the elderly position. Usually, the elderly persons are monitored with wearable sensor devices that communicate through a wireless sensor network (WSN) the medical data (such as pressure, heartbeat, etc.) to a server. In this work instead, we propose a distributed infrastructure where data can travel through shared buses and can be used by different components in order to infer critical situations. The key point of the proposed system is the availability of mechanisms, tools, and methodologies for the rapid prototyping of AAL applications built on top of it without knowledge about the network infrastructure or the communication protocols needed to communicate with the WSN.

We leverage the received signal strength (RSS) measured between the wearable sensors and the WSN to infer the patients position in the bed. Since the RSS does not require a special or a sophisticated hardware and it has become a standard feature in most of the wireless devices, the proposed technique is simple and minimally invasive. A wide variety of techniques and algorithms are found in the literature to classify measurements for posture and movement recognition. Most of them are based on traces collected using accelerometers and gyroscopes. Techniques range from feed-forward backpropagation neural networks [3] to discrete wavelet transforms [4], classification techniques [5–7], and hidden Markov models [8]. In this work support vector machine (SVM) classification techniques were implemented to recognize user's positions in the bed due to their success in many classification problems [5–7].

Our purpose is not to present a finely tuned and well-engineered algorithm, but to increase the performance in terms of energy consumption with respect to the state of the art. Moreover, in this work we will show that by exploiting the context information coming from the context bus of the proposed middleware platform, we are able to further decrease the overall energy consumption of the WSN.

The paper is organized as follows. Section 2 describes the reasons that led us to investigate this issue. Section 3 presents the middleware architecture and the bed position detection service. The SVM method used to classify the bed

positions is also briefly described. Results are reported and commented on in Section 4, and concluding remarks are drawn in Section 5.

2. Motivations

The use of a middleware infrastructure able to provide data from any kind of sensor installed at the assisted person's home is essential for AAL applications where context information can be shared among different services. The objective of this work is to provide a middleware infrastructure for the rapid prototyping of applications of ambient intelligence (AMI) for healthcare and AAL, with a certain degree of dependability. In particular, we propose a *bed position detection* service that is the input for other two important services, namely, *bedsores prevention* and *sleep monitoring* services. These services together are called *bed activity monitoring* (BAM) services.

The key point of the proposed work is the optimization, in terms of energy consumption, of the resources used by the services to infer useful information about the health, safety, and well-being of the assisted person. To do this, the use of context information coming from sensors and information about the activities of the user coming from dedicated activity recognition services becomes crucial to reduce the energy consumption of the overall WSN deployed in the environment. The proposed BAM services use a wearable device to monitor the sleeping activity of the user. Increasing the battery life of this mobile device means reducing the effort for recharging and maintenance of the device by the user, making him feel less in need of care.

2.1. AAL Middleware. Making software a *commodity* by developing an industry of reusable components was set as a goal in the early days of software engineering. Evolving access to information and to computing resources into a *utility*, like electric power or telecommunications, is also a big target in the current ICT research fields. While significant progress has been made towards these goals, their achievement still remains a long-term challenge. As stated in [9], the computing facilities of large enterprises are evolving into an utility. This is true for AAL system especially when integrated with Healthcare Information Systems (HIS), which should become far more portable from one site to another in order to limit development and maintenance costs. On the way to meeting this challenge, designers and developers of distributed software applications are confronted with more concrete problems in their day-to-day practice. Reusing legacy software, developing mediation systems, component-based architectures, or implementing client adaptation through proxies are situations where applications use intermediate software that resides on top of the operating systems, and communication protocols to perform the following functions: hiding distribution (i.e., the fact that an application is usually made up of many interconnected parts running in distributed locations), hiding heterogeneity (i.e., various hardware components, operating systems and communication protocols that are used by the different parts of an application), providing uniform and standard high-level interfaces (so that applications

can easily interoperate and be reused, ported, and composed), supplying a set of common services to perform various general-purpose functions, in order to avoid duplicating efforts and to facilitate collaboration between applications. This intermediate software layer have come to be known under the generic name of middleware. Using middleware has many benefits, most of which derive from abstraction: hiding low-level details, providing language and platform independence, reusing expertise and possibly code, easing application evolution. As a consequence, the application development cost is reduced, while quality (since most efforts may be devoted to application-specific problems), portability and interoperability are increased [10].

In this work, we propose a middleware that, despite its general-purpose nature, is well suited to the AAL context. Since the inner context-aware nature of AAL applications, the presence of a pervasive solution that provides any kind of information about the interaction between the user and the surrounding environment became a key aspect for their effectiveness.

2.2. Bed Activity Monitoring Services

2.2.1. Bed Position Detection Service. Continuous observations of the assisted person through a bed position detection service is necessary in order to prevent bedsores or to monitor his sleeping behaviour. This service is the core service for BAM services providing inputs for further analysis by other AAL services. The proposed component-based architecture lets developers use services in a modular way. Future services will be able to use the data produced by the bed position detection service without rewriting a new ad hoc component. The configurable parameters for this service are the observation time and the sampling frequency. By means of these two parameters, a more efficient use of the WSN deployed in the environment is possible.

2.2.2. Bedsores Prevention Service. Nursing home requires a caregiver that ideally observes the elderly around the clock to prevent bedsores. The caregivers have to provide a high degree of surveillance and attendance to the elderly all the time. Moreover, the knowledge and personality of caregivers affect the quality of nursing care. Lack of timely care and insufficient preventive measures taken by human caregivers leads to unfortunate consequences to be suffered by the elderly and also indirectly affects their family members. This can lead to further escalation in the already mounting health-care costs for the government and degradation of quality of life (QoL) for the elderly. The bedsores can be mainly caused due to unrelieved or constantly applied pressure over bony and bed sore-prone areas of the body. Bedsores are regarded as one of the serious diseases and take a long time to completely heal [11]. Figure 1 shows the areas of the body most vulnerable to bedsores: heels, hips, buttocks, and shoulders. In addition to the pain and embarrassment that accompanies bedsores, patients are at risk for developing a variety of medical complications such as sepsis, an infection of the blood from bacteria entering the body from a bed sore. The most widely accepted ways of preventing bedsores is to

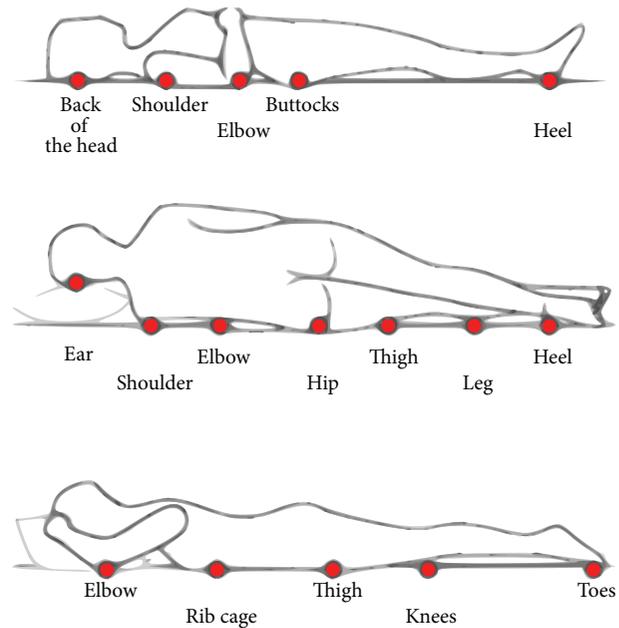


FIGURE 1: Diagram showing the areas of the body at risk of pressure sores when lying down.

actively turn the patients who have limited mobility on a regular basis (every 2 h) to avoid unrelieved pressure from forming on the body. Usually the caregivers use a turning sheet to keep track of the patient's position, recording the last position, the elapsed time, and the next position (turning plan).

In this work, we propose a service able to automatically assess the bed sore risk, able to help the caregiver to decide the care program and, thus, control that the actual patient's position matches with the turning plan, and increase the quality of nursing care. This is the scope of this service that wants to support the pressure ulcer prevention by (i) monitoring the patient's self-movements and adapting the caregivers interventions and (ii) decreasing the burden of the caregiver to prevent the bed sore. In fact, by knowing the posture of the subject, potential bed sore risks of the subject can be inferred and timely reminders can be sent to caregivers.

2.2.3. Sleep Monitoring Service. The sleep plays an important role in quality of life and is an important factor in staying healthy. Having inadequate and irregular sleeping patterns has a serious impact on our health and can lead to many serious diseases like cardiovascular disease, obesity, depression, and diabetes [12]. People can feel fatigued and cannot concentrate during the day if they do not sleep for a sufficient amount of time. This may be caused by interrupted sleep, such as having frequent periods of restlessness during sleep. Moreover, in many cases, particular body positions should be maintained or avoided (i.e., patients with obstructive sleep apnea should avoid sleeping supine [13]).

Sleep monitoring service is essential to recognize sleeping disorders for diagnosis and prompt treatment of disease. In

fact, it can provide healthcare providers with quantitative data about irregularity in sleeping periods and durations. Moreover, it can also provide detailed sleeping profiles that depict periods of restlessness and interruptions helping to find trends that correlate to certain diseases. Finally, it enables monitoring effectiveness of treatments to sleep-related diseases. Many studies (such as [14]) are focused on finding correlations between body positions during sleep and various breathing problems (e.g., sleep apnea). Therefore, a bed position detection service that provides information about body positions during sleep helps such type of studies.

3. The Proposed Solution

The goal of our work is to infer the elderly position in the bed, without using an ad hoc or sophisticated hardware. In fact, we suppose that the elderly/patient wear any wireless sensor device able to transmit (hereafter also called mobile), and that the environment is equipped with fixed wireless devices (hereafter also called anchors) installed transparently at home in general-purpose devices such as lights, mains power outlets, or light switches. Indeed, recent advances in sensor technology have enabled the development and deployment of wearable systems for patients' remote monitoring. We envision that in the near future wearable sensor will be placed directly on the woven in order to increase the acceptability of the patient. In [15], the authors describe the key enabling technologies to monitor the patients using acceptable wearable sensors such as flexible wireless sensor and e-textile technology.

In this work, we propose a bed position detection service that, leveraging the RSS measured between the mobile and the anchors, infers the patient's bed position. The information about the position while the patient is in bed would allow us to support caregivers in the prevention of bedsores and monitoring of patient's sleeping behaviour. Indeed, the proposed services would be able to alert the caregivers if the position of the patient keeps fixed for a long time, to tailoring the interventions to the patient's current needs (bedsores prevention service) or to recognize sleeping disorders as early as possible for diagnosis and prompt treatment of disease (sleep monitoring service).

In the following, we will describe the proposed middleware architecture, the proposed bed position detection service (that, as already highlighted in Section 2, is the main building block of any bed activity monitoring services), and, finally, the practical usage of the overall system.

3.1. Middleware Architecture. The sensors, the services, and the components integrated in the system will use a software infrastructure, which is based on a middleware that hides heterogeneity and distribution of the computational resources in the environment. The proposed middleware solution uses a Java/OSGi platform as the reference platform for the development. However, the integration of such components is demanding, especially if we consider that the system is composed of different services written in different languages and it may need to be accessible by a number of remote healthcare centers which may use different protocols. The fragmentation

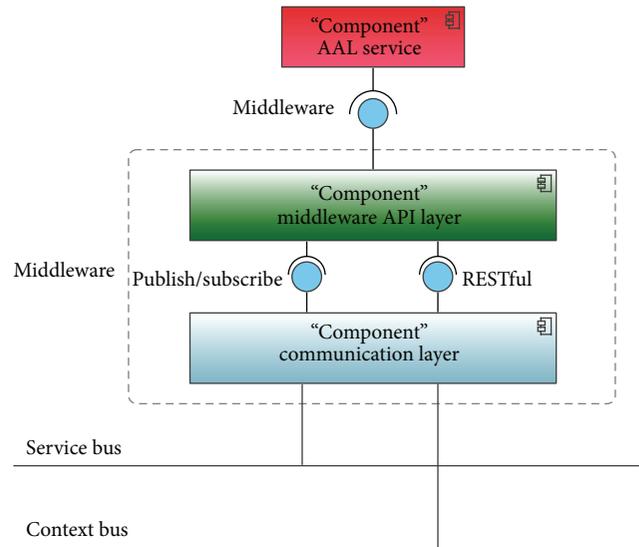


FIGURE 2: The proposed middleware architecture.

in this sector is still high, but there are initiatives working to build converging solutions. Service interoperability is a key point to build an ecosystem of applications that helps the growth of an ambient assisted living (AAL) consumer market. In this regards, several European projects have intensely worked on the definition and standardization of a common platform for AAL, on top of which to develop intelligent software applications for the end users. Our objective is to design and develop a system compliant to the results of the most promising research projects in this field [16]. Within the proposed AAL ecosystem, an AAL space is intended to be the physical environment—such as the home of an assisted person—in which independent living services are provided to people that need any sort of assistance. In such a virtual ecosystem, hardware as well as software components can “live” while being able to share their capabilities. In this space, the proposed platform facilitates the sharing of two types of capabilities: service (description, discovery, and control of components) and context (data based on shared models). Therefore, connecting components to the platform is equivalent to using the brokerage mechanism of the middleware in these two areas for interacting with other components in the system. Such connectors together with the application logic behind the connected component are called altogether “AAL services.”

3.1.1. Middleware Layers. The concrete middleware architecture is made up of two layers: a core middleware API layer and a communication layer that includes a publish/subscribe connector and a RESTful connector (Figure 2). A generic service built upon the middleware can discover which sensors are present in the environment and other services together with their functionalities using methods from the middleware API layer. The underlying layer fulfils these requests exploiting the connectors available. In the communication layer, an MQTT [17] and a RESTful [18] connector are present.

By means of these connectors, the middleware realizes a publish/subscribe and a method description and invocation mechanism transparently to the services that use them.

3.1.2. The Buses. Two buses form the heart of the proposed middleware: a context bus and a service bus. All communications between applications can happen in a round-about way via one of them, even if physically, the applications are located on the same hardware node. Each of the buses handles a specific type of message/request and is realized by different kinds of topics. The aim of the middleware is to provide a publish/subscribe mechanism for accessing the context information about the physical environment and physiological data. This information will be exposed as different topics: topics for discovery and description of devices and services that form the service bus and topics for publishing and retrieving data from devices and services that form the context bus.

3.1.3. The Announce Mechanism. The middleware is in charge of presenting the available sensors and services in the system implementing an announce mechanism on the service bus. A generic resource (i.e., a sensors exporter or a service) uses this mechanism to notify its subscribed system components the presence and modification of the exported resources. This process of announcing and exchanging available content is efficiently implemented using a particular message on the relative topic in the service bus. The message is a descriptor file containing an id, a description, a type (i.e., exporter or service), a set of resources (i.e., sensors or components), and a set of methods. Once a resource has been announced on the service bus, a generic service can search for it filtering the descriptor fields and use it. A generic service can also subscribe to future available resources or particular kind of them by means of filters. When the required resource becomes available, the subscribers will be notified with the corresponding descriptor. In the same way when a resource is modified or becomes unavailable, the subscribed services will be notified with the new descriptor in case of modification or null value in case of unavailability. The topic used for announcement and discovery of devices and services, the so-called service bus, has this format:

```
<<location>>\serviceBus\<<serviceID>> ,
```

where location identifies the room in the assisted person's apartment, serviceBus is the keyword to identify the topic as a service bus topic, and serviceID is the unique identifier of the service. The message of this topic is a JavaScript Object Notation (JSON) [19] descriptor file. JSON is a lightweight-text-based open standard for client/server data exchange, and an opportunity in the Internet of Things (IoT) [20] ecosystem arises from the choice of JSON: connected sensor nodes can use standard Internet protocols with lightweight web services based on REST/HTTP. In the case of sensors with low-resource hardware, the exporter, generally installed in a gateway node with higher hardware specifications, is in charge of translating small data packets from sensor motes into JSON descriptors.

3.1.4. The Context Bus. The middleware takes care of dispatching information about the state of the resources among services by means of a context bus. Any service that wants to make his data available (sensors readings and events or data analysis results) can use the middleware API to publish it. Any service interested in monitoring these data can subscribe to the relative context bus topics indicated in the descriptor using the middleware API. The topic used for gathering data from devices and service, the so-called context bus, has this format:

```
<<location>>\contextBus\<<serviceID>>\
<<subtreefield>> ,
```

where location identifies the room, contextBus is the keyword to identify the topic as a context bus topic, serviceID is the unique identifier of the service, and the subtreefield identifies all the resources of that service that can be monitored. For each resource there will be a dedicated context bus subtopic. The message of these topics is a string value.

3.1.5. Overall System Architecture. Figure 3 shows the proposed system architecture where a particular kind of AAL services, called exporters, provides data from a generic WSN to other services in the environment. Different exporters can be present to let services interact with different WSN technologies (i.e., ZigBee, Bluetooth, and KNX). The exporter acts like a gateway for the WSN. It announces on the service bus the presence of a new device installed and publishes events and sensed data on the context bus. It can be also configured using the API described in its descriptor file. Another important service of the proposed system is the simple activity recognition service. It is in charge of publish simple activity information like "is sleeping," "is cooking," and "is bathing" based on the context data gathered from the context bus. In the proposed scenario, data from a presence pressure detection sensor, a passive infrared sensor (PIR), and a light sensor are used to infer the activity "is sleeping." The core components of the proposed system are the bed activity monitoring (BAM) services including a bed position detection, a bedsores prevention, and a sleep monitoring service. These services gather RSS sensors data and activity data from the context bus in order to monitor context and behaviour of subjects on the bed. The bed activities events and alerts will be published, and a remote interoperability service can transmit and visualize them to remote relatives, telecare centers, and hospitals.

3.2. Bed Position Detection Service

3.2.1. Devices. In order to retrieve environmental data and infer simple activities, we test our system by using wireless sensors provided by different manufacturers. The bed occupancy sensor is produced by Tunstall [21] and consists of a bed pressure pad, which is placed underneath the mattress of the user. The sensor raises events when the user is in or out the bed transmitting on the dedicated European 869 MHz social alarm frequency. The same frequency is used by Tunstall's

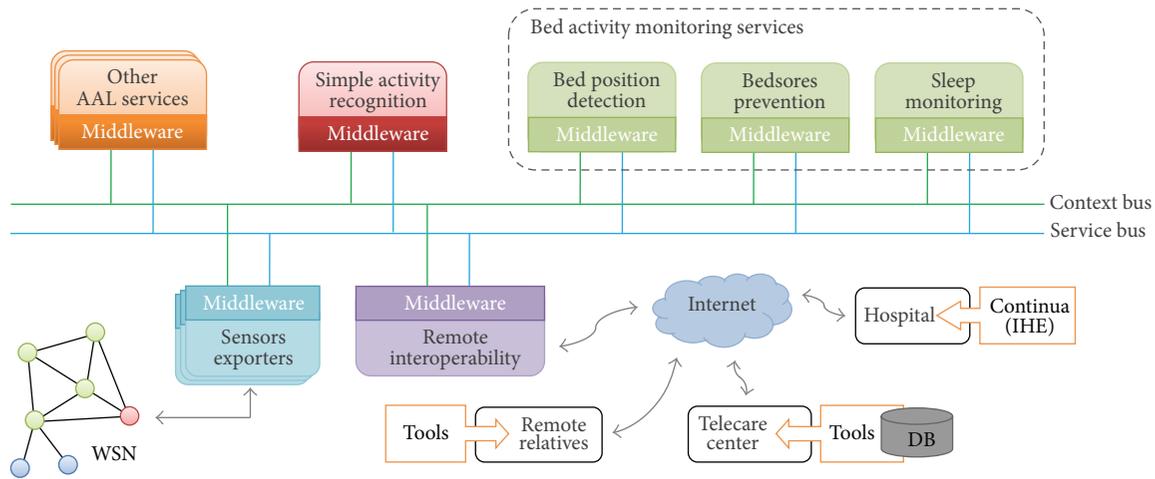


FIGURE 3: The proposed system architecture.



FIGURE 4: The sensors used in the test system: (a) bed occupancy sensor, (b) light sensor, (c) PIR, and (d) IRIS transceivers.

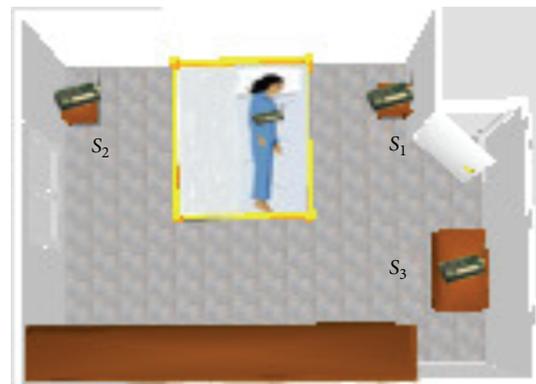


FIGURE 5: Setup environment: three fixed sensors S_1 , S_2 , and S_3 are placed on the environment. S_1 is placed on the nightstand, S_2 is placed on the dresser, and S_3 is placed on the wardrobe. The user wears a mobile sensor placed on the breast.

wireless PIR detectors delivering motion detection. The sensors are connected to a gateway that collects the events raised. A middleware instance is installed on the gateway node to announce sensor descriptors on the service bus and the events on the context bus. Also an illuminance sensor which contains a light sensor that detects the level of light in a room has been used. The sensor, called ZLum, is a wireless product produced by Cleode [22] that sends its alarms on ZigBee network. It is compliant with ZigBee Pro 2007 stack. A ZigBee gateway has been used and integrated in the middleware. In order to investigate how the RSS measured between wireless devices can be used to infer the elderly position in the bed, we test our system by using a WSN composed of Crossbow IRIS transceivers [23] operating at 2.4 GHz (ISM band) according to the IEEE 802.15.4 protocol [24]. The sensors include an Atmel ATmega1281 microcontroller, 128 KiB flash memory to store the executable code, 512 KiB serial (slow) flash memory to store data, and 8 KiB RAM. The transceiver is powered by two AA batteries and draws 8 mA in active mode plus 17 mA in continuous Tx mode at max power (3 dBm) and 20 mA in Rx mode. The antenna is a 1/4 wave monopole. Figure 4 shows the sensors used in the test system.

3.2.2. Setup and Integration. The environment chosen to test the proposed solution is a typically bedroom with wardrobe, nightstands, and a dresser. Three fixed sensors are placed on the environment as highlighted in Figure 5; two sensors were placed at about 55 cm height (S_1 and S_2), while the last one (namely S_3) was placed on the dresser at about 85 cm height. The users wear a mobile sensor that was placed on the breast. The bed positions we take into account in this work are summarized in Table 1. We conducted a series of experiments (fifty) that consist of cyclical repeat in all the bed positions. The user's positions are held at least 10 seconds, and the fifty repetitions were performed in different days to verify the experiment repeatability. The *sampling frequency* should be chosen considering on one hand the computing constraints and networking overhead, which are both directly responsible for power consumption in the sensors, and on the other hand RSS waveform reconstruction accuracy. In this work, we evaluate the accuracy of the bed position detection service collecting the RSS measures for an *observation period* of 2 seconds and with a sampling frequency of 1, 2, 4 and

TABLE 1: Schematic representation of the considered positions.

Prone—position 1	Left lateral—position 2	Supine—position 3	Right lateral—position 4
			
Lie on their bellies with head turned to the side	Sleep on left side with both arms down	Lie on their backs with arms up or down	Sleep on right side with both arms down

```

{
  "id": "crossbowIris",
  "type": "exporter",
  "description": "The exporter for Crossbow Iris sensors",
  "resources":
  [
    {"deviceId": "Anchor1", "freq": "1"},
    {"deviceId": "Anchor2", "freq": "1"},
    {"deviceId": "Anchor3", "freq": "1"},
    {"deviceId": "Body1", "freq": "1"}
  ],
  "getSamplingFrequency":
  {
    "type": "method",
    "description": "GET method responsible for obtaining the sampling frequency of the transceivers",
    "returns": "number",
    "params": ""
  },
  "setSamplingFrequency":
  {
    "type": "method",
    "description": "SET method responsible for obtaining the sampling frequency of the transceivers",
    "returns": "boolean",
    "params": {"deviceId": {"type": "number"}, "freq": {"type": "number"}}
  },
  ...
}

```

PSEUDOCODE 1: The crossbow exporter JSON descriptor file.

8 Hz. The time between two consecutive observations is a parameter that is set to zero in order to generate the user's position every observation period. The Crossbow sensors have been integrated in the system through an OSGi bundle

(the exporter) that wraps the procedures of the embedded operating system installed on the transceivers. Pseudocode 1 shows a snippet of the message published on the service bus once the exporter is started. The descriptor file of the bed position detection service will present similar methods to get and set the observation period and the sample frequency.

3.2.3. The Proposed Method. In order to infer the user's position, we used the well-known SVM classification method [25, 26]. In our case, each bedside position produced three traces (one for each receiver, S_1 , S_2 , and S_3); each triple-trace is an object to be classified. These triple-traces were identified by up to four features, as described in the next section. The specific features extracted from the RSS traces were chosen using Weka, a collection of tools for data preprocessing, classification, clustering and more [27]. Weka was also used to evaluate the performance of the SVM algorithm [25, 26]. A one-against-one approach was used to tackle the multiclass classification problem. The classification is made by a max-wins voting strategy. This method constructs a classifier that is trained for every pair of classes (in our case a class was associated to a specific position). After that, for a given test sequence, each classifier assigns one vote, and the object is assigned to the class with the highest number of votes. For both classification methods, classification performance was computed by using a 10-fold cross-validation technique; that is, an object set (a triplet of traces, for each position) was randomly subdivided into 10 equal-sized partitions: 9 of them were used as the training dataset and the last one was used as the testing dataset. The same procedure was repeated 10 times, until each partition was used for testing. In this way, each object was used exactly once for testing.

3.3. Practical Usage. In order to clarify the overall functionalities of the proposed system, a simple usage scenario will be shown (Figure 7). After an installation phase, any device able to communicate its state present at home will be available on the service bus and will publish sensed data on the context bus. We want to monitor the sleeping behaviours and critical situations when the assisted person is in bed. For this reason,

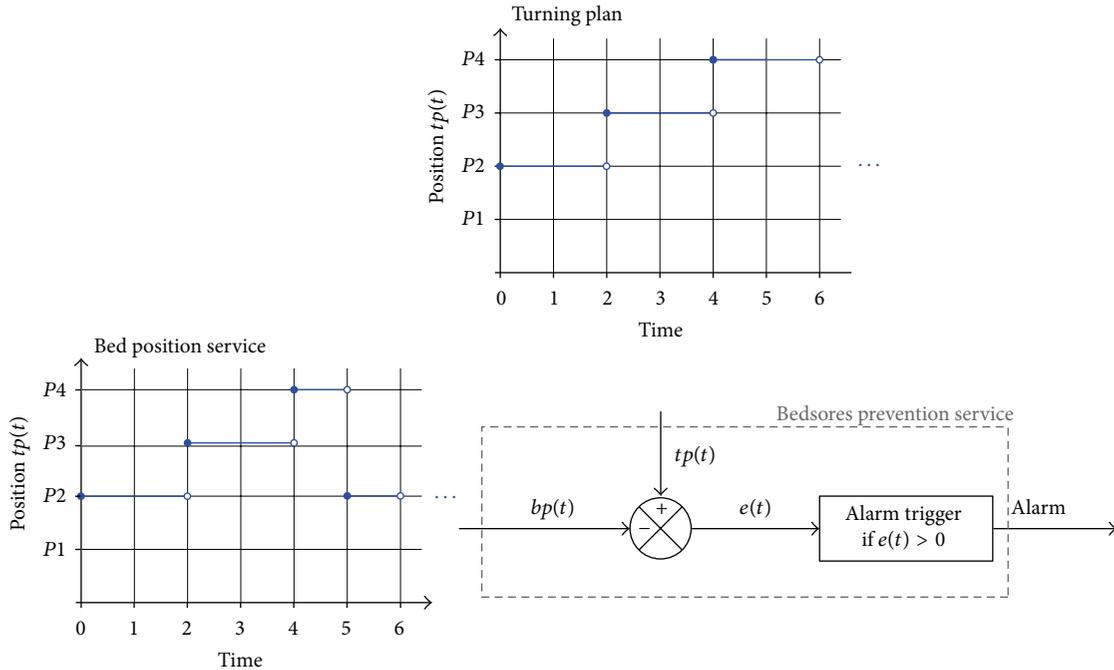


FIGURE 6: The bedsore detection service reasoning. If the actual position $bp(t)$ of the patient does not match with the turning plan position $tp(t)$, an alarm is raised.

the simple activity recognition service will be listening to the sensed data coming from the PIR, the bedpresence pressure detection sensor, and from the light sensors. When the assisted person goes to sleep, first the PIR indicates that there is a movement in the bedroom, then the bedpresence will communicate that there is somebody in the bed, and then the light sensor will publish data about the change of illuminance level in the room. Collecting these context data, the simple activity detection service infers that the person is going to sleep and will publish the “isSleeping” activity event on the context bus. The bed activity monitoring services are listening for this event because they are subscribers for that topic. Depending on the type of BAM service selected by the caregiver, once the event is received, the chosen service will configure the bed position detection service in terms of observation period and sampling frequency in order to optimize the resources utilization. Then, it will start the bed position detection service and begin to collect bed position events. For each patient, the proposed system must be trained, collecting the RSS traces in each position. After the training phase, the bedsore prevention service will publish alarms if a critical bedsore situation is detected. Since the patients need to be moved every two hours in order to prevent the risk of bedsore, we propose a service able to control that the actual position of the patient matches with the turning plan. Therefore, the service will publish alarms if the position of the patient (detected by our system) and the turning plan does not match (Figure 6).

If the sleeping monitoring service is running, it will report the collected bed positions timeline. The bed positions timeline will be accessible via the RESTful API described in the relative descriptor file on the service bus. Once

the assisted person wake up, the simple activity recognition service will publish the event “isAwake” to the context bus and the services listening for this event will stop their activities. Figure 7 shows the activity diagram for the bedsore prevention service. In our experimental setup, the wireless sensor devices were close to each other (in the same room); this deployment ensured a negligible packet loss. Indeed, during our measurement campaign we did not observe any packet loss.

4. Results and Discussions

In this section, the results of the bed position detection service are presented. In particular, we evaluated the performance of the proposed method in terms of accuracy, energy consumption, and responsiveness.

4.1. Preliminary RSS Traces Analysis. Figure 8 presents an example of typical RSS 40-minute registrations for the bed positions. The variations between RSS traces relevant to different user’s position are clearly apparent. In particular, the RSS values when S_3 is used for the left and right lateral positions are quite similar, as well as the prone and supine positions for S_2 . For this reason, exploiting more sensors and/or more RSS features, the classification performance should increase. In the following, we will describe the extracted RSS features showing also the achieved performance.

4.2. Feature Extraction. The first step of the classification procedure was to identify a limited number of features that act as the “fingerprint” of a trace. An initial large set of

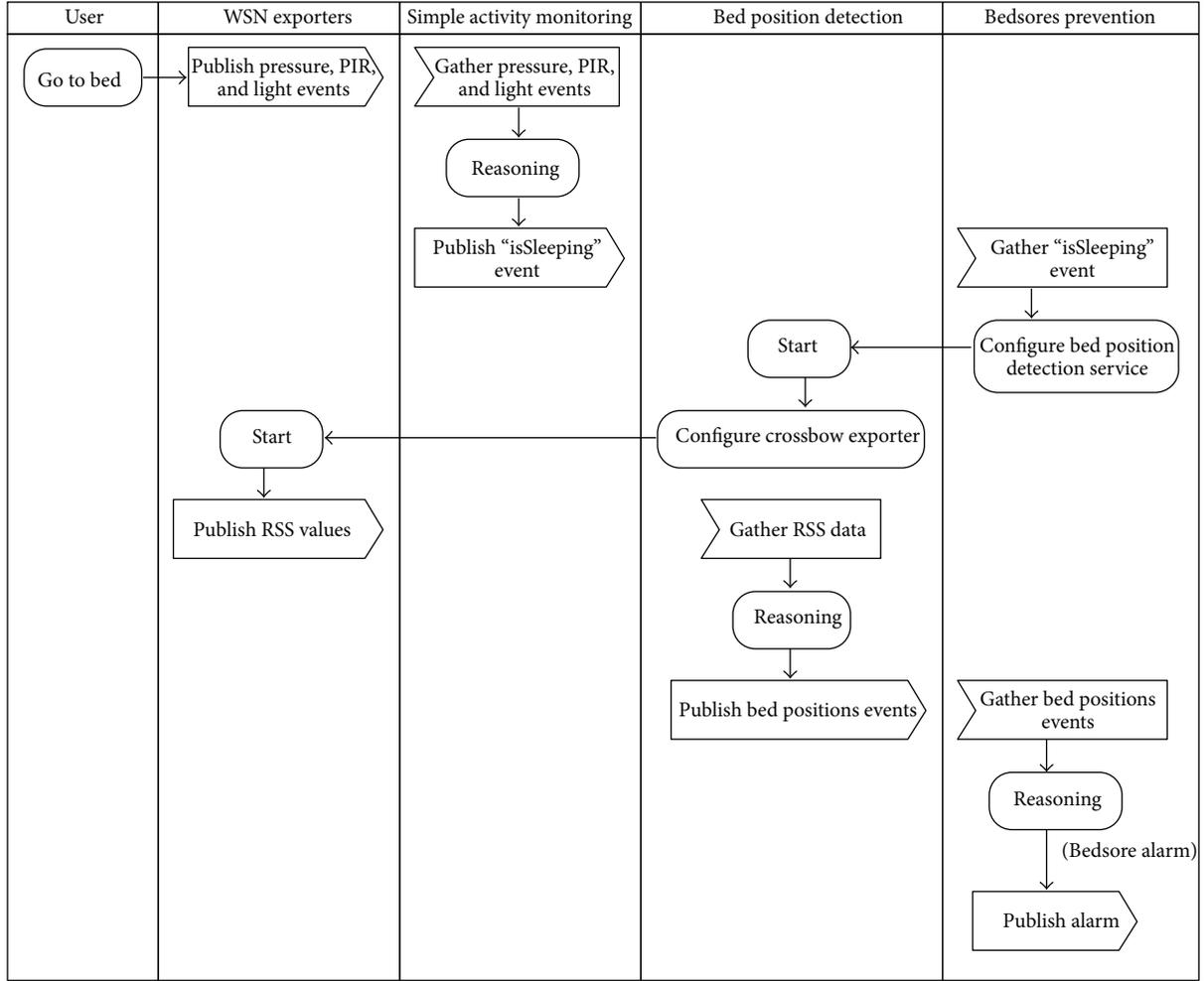


FIGURE 7: The activity diagram of the bedsore prevention usage scenario.

possible features was defined, from which the best performers were chosen using the feature selection tools provided by Weka. In the set of possible features, we considered both time-independent and time-series-based statistics. As far as time-independent statistics are concerned, the ones involving only one transceiver (either S_1 , S_2 , or S_3) were mean value μ , standard deviation σ , skewness, and kurtosis. The one involving two transceivers (chosen among S_1 , S_2 , and S_3) was the cross-correlation ρ . As far as time-series-based statistics are concerned, we considered the level crossing rate (LCR) at four different thresholds, firstly computed on each device separately, and secondly on the difference of the devices' RSS measurements. The LCR is a statistical parameter that quantifies how often the signal crosses a given threshold in the positive-going direction. The four thresholds considered in this work were LCR1 at $\mu - 0.5\sigma$, LCR2 at $\mu + 0.5\sigma$, LCR3 at $\mu - \sigma$, and LCR4 at $\mu + \sigma$, since they are the most used LCR thresholds used in the literature [6, 7, 24]. A features' short list was selected from the initial large set data in order to optimize classification performance. If two of the three transceivers are used, the list of features includes two mean values among μ_1 , μ_2 , or μ_3 and two standard deviations among σ_1 , σ_2 , or

σ_3 . If only one sensor is used, the feature list includes the mean value μ , the standard deviation σ , LCR2, and LCR4. An example of how some features are distributed, changing the number of exploited sensors is shown in Figure 9. As shown in Figure 9(a), exploiting only one sensor and hence LCR2, μ , and σ , Positions 1 and 3 are well separated from Positions 2 and 4, which means that Positions 1 and 3 can be well recognized, whereas Positions 2 and 4 are more difficult to identify and may be confused to each other. Instead, when we chose to exploit all the three sensors and μ_1 , μ_2 , and μ_3 , Figure 9(b) shows that all the positions can be easily recognized since the features are well distributed.

4.3. *Experimental Results.* Performance of the proposed system is measured in terms of error rate or, equivalently, of matching rate (i.e., its complementary) and in terms of true positive rate (TPR) and false positive rate (FPR) measures defined as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad (1)$$

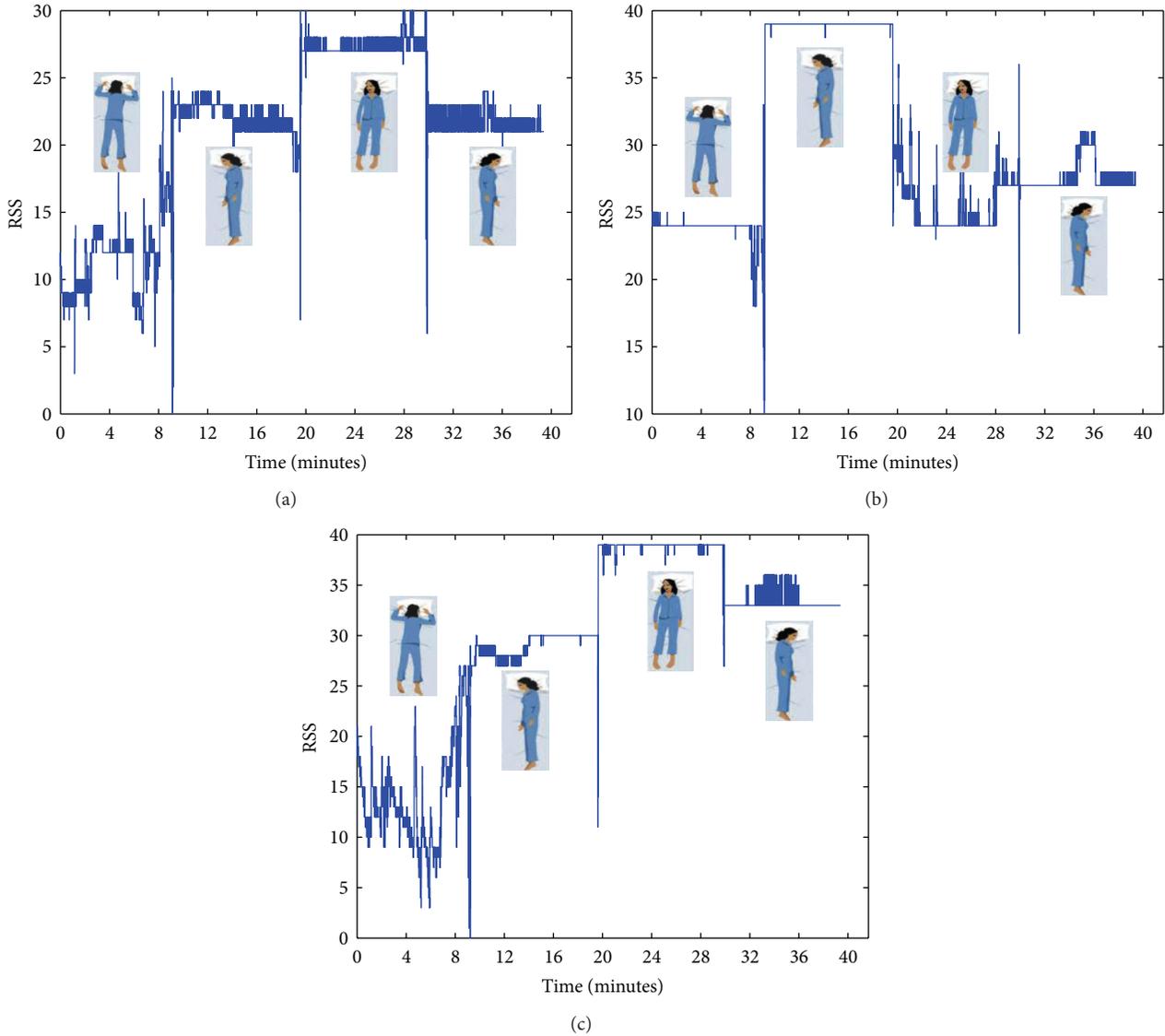


FIGURE 8: Samples of RSS traces of the five different bed positions estimated from sensors (a) S_1 , (b) S_2 , and (c) S_3 .

where true positive (TP) test result is the one that detects the condition when the condition is present, a false negative (FN) is defined as a negative result on a test when the condition is present, a false positive (FP) test result is the one that detects the condition when the condition is absent, and a true negative (TN) test result is the one that does not detect the condition when the condition is actually absent.

Figure 10(a) shows the error rate using only sensor S_2 as a function of the number of features, when the sampling rate was 1, 2, 4, or 8 Hz. We chose the sensor S_2 since it did not get neither the best nor worst performances. Firstly, one feature was considered (μ_2) achieving about 88% of matching rate at 1 Hz. The matching rate increases with the number of features, as expected. In fact, when using two features (μ_2, σ_2), 94% matching rates were achieved. In this case, the use of LCR2 and LCR4 does not significantly improve the performance.

Moreover, increasing the sample frequency, the performance in terms of error rate shown in Figure 10(a) increases.

Figure 10(b) shows the TP and the FP rates, considering sensor S_2 and only two features (μ_2 and σ_2). Positions 1 and 2 exhibited 100% TP and 0% FP, while Position 3 was classified with 75% TP and 0% FP and Position 4 was classified with 100% TP and 8% FP, when the sample frequency of 1 Hz was chosen. Position 4 (right lateral) presented the highest value of FP, which means that it was the most often misclassified one. In fact, as we will see later Position 3 and Position 4 are misclassified; that is, Position 4 is classified as 3 and vice versa. Moreover, Position 1 (prone) and Position 2 (left lateral) had the highest values of TP and the lowest value of FP, making them the most correctly recognized movements. When the SVM algorithm was used, Positions 1, 2, and 4 were the most correctly recognized movements, while Position 3 was

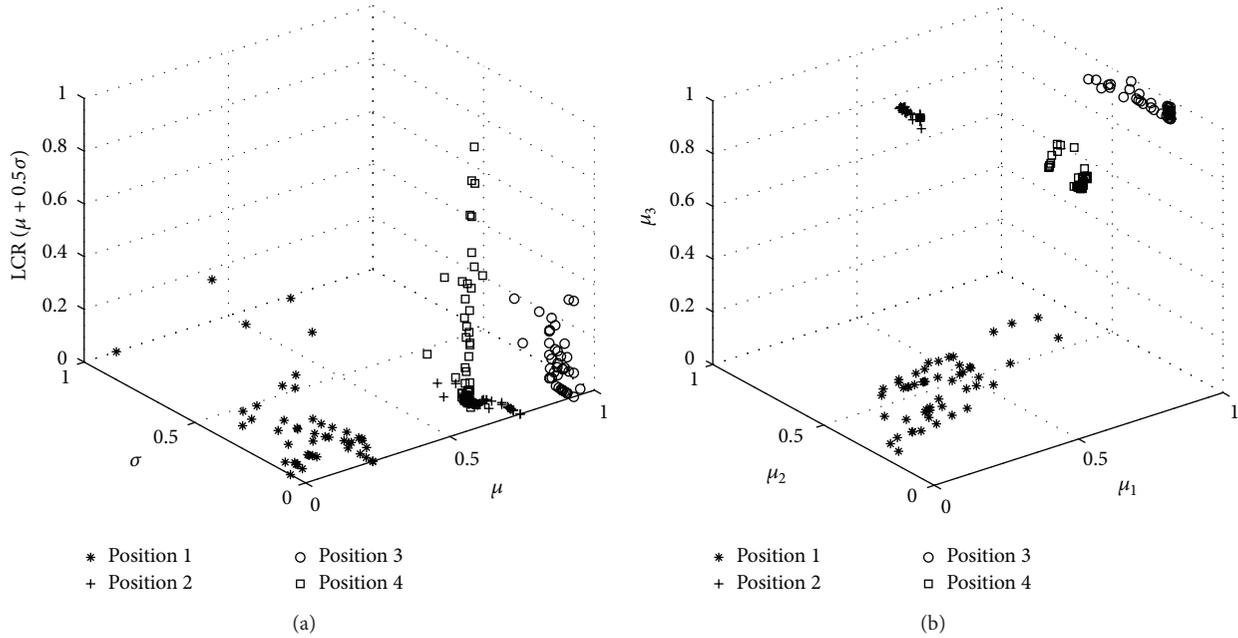


FIGURE 9: Two examples representing objects in their feature space at 8 Hz (a) using S_3 and hence μ_3 , σ_3 , and $LCR2_3$ and (b) using three sensors and μ_1 , μ_2 , or μ_3 .

the most often misclassified one. Figure 10(b) shows also how the performance increases increasing the sample frequency. In fact, at 4 Hz Position 3 was classified with 92% TP and 0.5% FP and Position 4 with 98% TP and 2% FP.

In order to evaluate which transceiver performs better, the performance of the SVM algorithm is shown. Figure 10(c) shows the classification error rate at 4 Hz, for each sensor S_1 , S_2 , and S_3 as a function of the number of features. The features considered were μ , σ , $LCR2$, and $LCR4$ for each sensor. Sensor S_1 exhibits better performance, since it experienced greater RSS variations with respect to the other sensors, and hence the algorithm is able to better distinguish the various positions. Two features (μ and σ) are sufficient to achieve a low error rate; the use of $LCR2$ and $LCR4$ does not further increase the performance. Figure 10(d) shows TP and FP rates at 4 Hz using two features (μ , σ) of sensors S_1 , S_2 , and S_3 . Positions 2 and 3 were always recognized (100% TP) and were rarely confused with other positions by leveraging the RSS from S_2 or S_1 , respectively. Position 1 shows 100% TP and 0% FP when using S_2 , while Position 4 was classified with 98% TP, and 0.5% and 2% FP by using S_1 and S_2 , respectively.

Concluding, using only one sensor device, the better performance is achieved by using the sensor S_1 , probably because it is more in line of sight region (LOS) with respect to the other cases and it is more close to the mobile device. If we want to reach a 100% of matching rate for all the considered positions, the conducted experiments show that the RSS traces of at least two of three sensors should be exploited together.

Finally, confusion matrices for the analyzed classification problem at 1 Hz, when the RSS traces of S_2 are exploited, are presented in Figure 11. Confusion matrices are a compact

graphical representation where each row of the matrix corresponds to the position assigned by the classifier (predicted class), while each column represents the position performed (actual class). A classification method with ideal performance will only have bars on the main diagonal of the matrix. The more bars on the nondiagonal cells are high, the worst the classification performance is. As far as the one feature case (μ_2) is concerned, there was confusion between Position 3 and Position 4. In fact, 46% Position 3 (supine) was misclassified as Position 4 (right lateral), while Position 4 was never misclassified as Position 2. On the other hand, Positions 1 and 3 were well recognized (100% of matching rate) even with one feature only. As expected, the greater the number of features, the less the error rate, except for Position 3 that also using four features was misclassified in the 25% of cases as Position 4. However, the performance does not change much between using 2 or 4 features. SVM method has the same behaviour when the sample frequency increases, that is, the greater the sample frequency the less the error rate. In fact, at 8 Hz Position 3 (exploiting only two features) was misclassified in the 3% of cases Position 4 and vice versa. This behaviour is due to the location of sensor 2 that was near to the left corner. Sensor 1 performs better with respect to the other sensors, in fact by leveraging 2 features it achieved the 99% matching rate (Figure 10(c)).

Summarizing, by using the RSS traces measured by only one well-deployed sensor (S_1) at 1 Hz with an observation period of 2 seconds, 99% matching rates were achieved with only two features (μ and σ). There are at least three factors that influence the matching rates: the position of the sensors, the sample frequency, and the observation period. The sensor must be placed much close as possible to the mobile

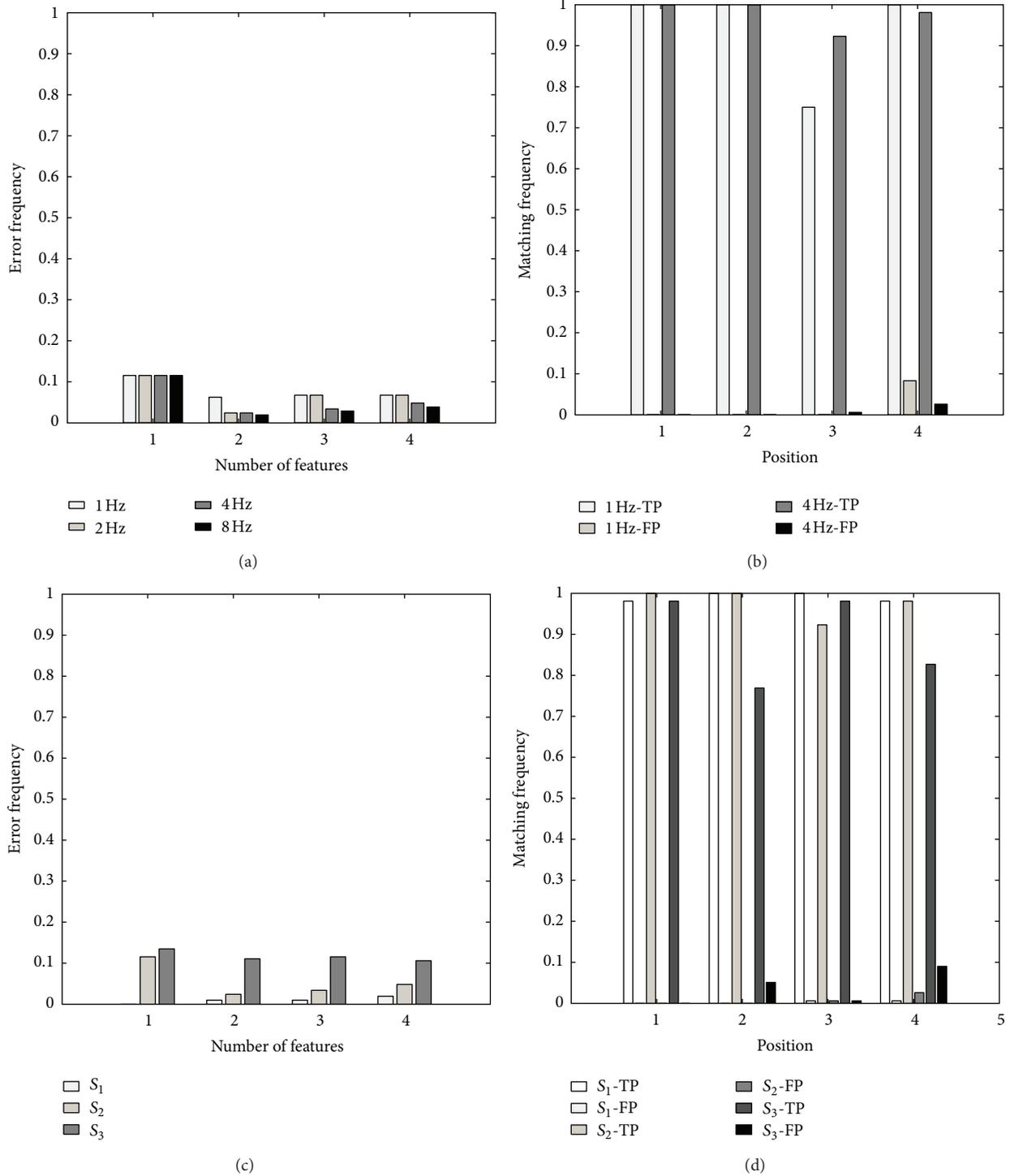


FIGURE 10: Classification performance using S_2 and the SVM algorithm: (a) shows the error rate as a function of the number of features of sensor S_2 , evaluated at 1, 2, 4, and 8 Hz; the features considered were μ_2 , σ_2 , LCR2, and LCR4, (b) shows true positive and false positive Rates for each bed position, two features of the sensor S_2 (μ_2 and σ_2), (c) shows the error rate as a function of the number of features for each sensor at 4 Hz, and (d) shows true positive and false positive rates at 4 Hz for each bed position for each sensor, two features (μ , σ).

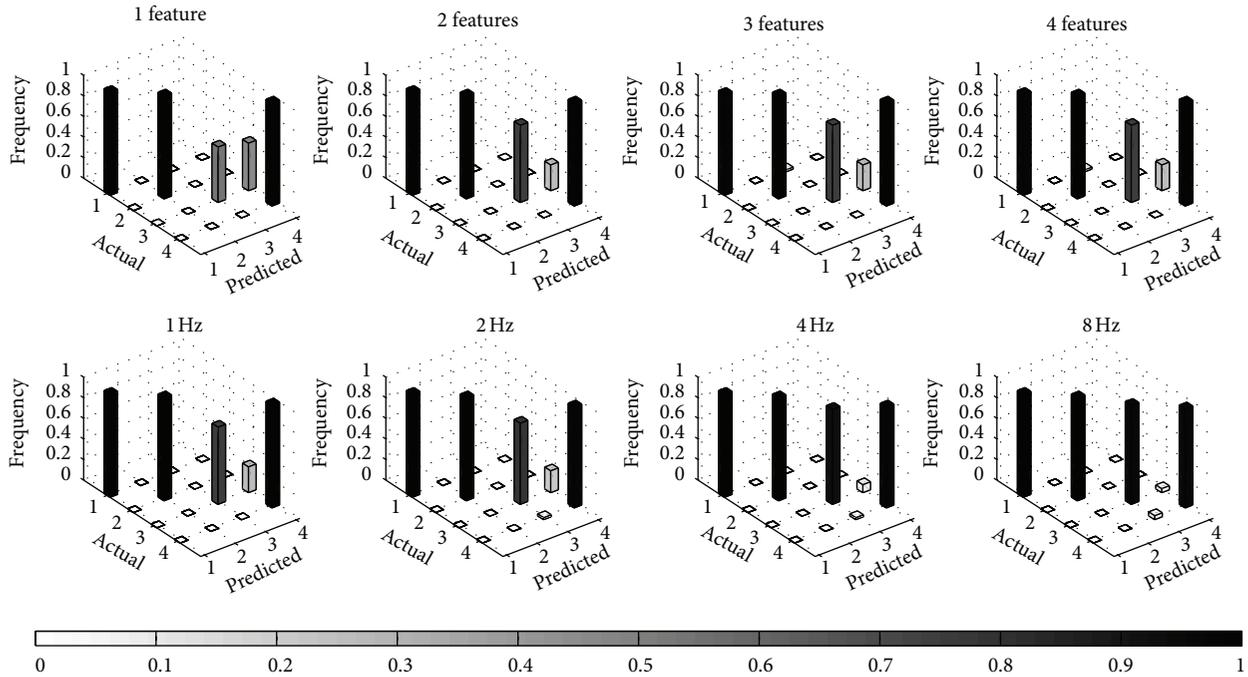


FIGURE 11: Confusion matrices for the bed positions. The two axes on the base of each graph represent the actual position class and the class predicted by the algorithm, respectively. The performance achieved by leveraging S_2 is given as a function of the features number (μ , σ , LCR2, and LCR4) at 1 Hz, and sample frequency (1, 2, 4, and 8 Hz) when only two features (μ and σ) are selected. The smaller the bars outside of the main diagonal, the better the performance.

(more in LOS) in order to have a stable signal over time. Lower value of sampling frequency and observation period implies a lower value of energy consumption and a service more responsive but with a degradation of the accuracy. In the next section, we will analyse the best communication architecture in terms of mobile lifetime, and we will show how decreasing the observation period fixes both the energy consumption and the accuracy performance.

4.4. Energy Consumption. As already discussed in Section 2, energy efficiency is recognised as a paramount property of any mobile device, in particular for WSNs. Consequently, the proposed BAM services must be designed to minimize the power consumption, especially if the services run for long periods. There are two different approaches to measure the RSS between mobile and anchors. In the first, the mobile receives the beacon emitted from each anchor, it computes the corresponding RSS, and it publishes these values to the context bus, while in the second approach the anchors receive the broadcast beacons emitted from the mobile, and later each anchor transmits the measured RSS to the context bus. As already discussed in Section 3.2.1, the transceiver draws 17 mA and 20 mA in transmitting and receiving modes, respectively. An analysis of the energy consumption of the two different approaches is reported in Figure 12. In fact, Figure 12 shows the lifetime of the mobile as a function of the sampling rate of the wireless channel, assuming that the sensor is a micaZ mote [23] equipped with an IEEE 802.15.4 radio subsystem and a battery of 2000 mAh. We can note that when mobile node transmit, using 1 Hz sampling rate

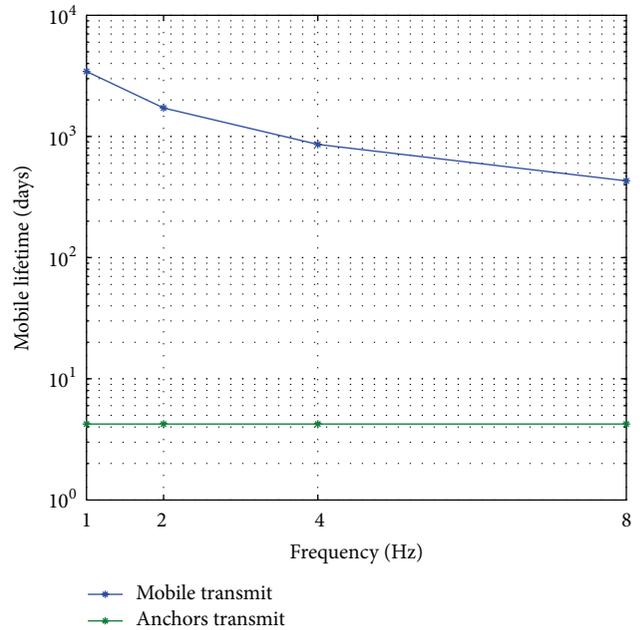


FIGURE 12: Mobile sensor lifetime considering a battery of 2000 mAh.

triplicates, the battery life compared to a 8 Hz sampling rate. It is worth to note also that if the mobile device is deployed on a network where other applications exist, the system can exploits the communications required by the applications

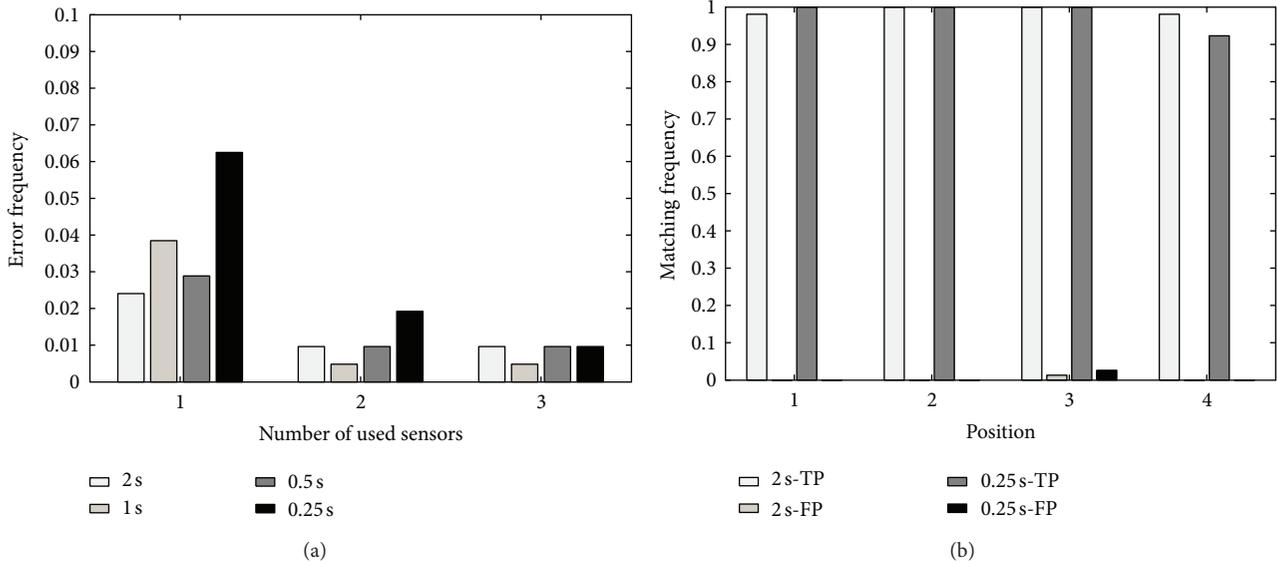


FIGURE 13: Classification performance exploiting two features (μ and σ) and fixing the number of the emitted beacon (hence fixing the energy consumption) to 2: (a) shows the error rate as a function of the number of used sensors, evaluated fixing the observation period to 2, 1, 0.5, and 0.25 seconds; (b) shows true positive and false positive rates for each bed position, two used sensors (S_1 and S_2).

themselves to sample the channel, thus reducing the actual number of beacons sent by the mobile node. An important improvement to the energy consumption reduction is given also by the use of the context awareness provided by the proposed overall system. Thanks to the context information shared between the services installed on the middleware, it is possible to infer simple activity of the user. In our scenario, the continuous monitoring of the bed activity is useless if nobody is sleeping. For this reason a simple activity recognition service is present. In this way, the WSN is used only when needed, especially the mobile nodes. Since the mobile sensor consumes more energy receiving the beacons instead of transmitting, the approach that allows to decrease the energy consumption is the second one, where the mobile emits the beacons and the anchors measure the RSS values (Figure 12). Indeed, when the anchors transmit, the mobile must receive a number of packets equal to the number of anchors deployed in the environment (that means consuming energy in receiving mode proportionally to the number of anchors) and later transmit a packet that contain the measured RSS values to the bed position detection service. While when the mobile transmits the beacon, the tasks of the anchors are receive the beacon, measure the RSS, and send this value to the bed position detection service. Furthermore, since the mobile only has to send a beacon periodically, it can remain in idle mode between any two consecutive beacons (which means that, between any two beacons, it can turn off the radio and the processing subsystems).

4.5. Observation Period. In this section, we want to analyse, fixing the energy consumption (i.e., number of emitted beacons), the accuracy performance of the proposed SVM method decreasing the observation period. This parameter need to be as lower as possible for real-time services while it

could be higher otherwise. Figure 13 shows the performance achieved by the proposed SVM method, fixing the number of emitted beacons to 2, that corresponds to a mobile lifetime of about 3000 days. As depicted in Figure 13(a), increasing the number of exploited sensors, the error rate decreases. While decreasing the observation period from 2 to 0.25 seconds, the error rate increases from 2.5% to 6.3% when only the sensor S_2 is used. However, the error rate decreases to less than 1% when three sensors are used for any value of observation period. Finally, Figure 13(b) shows the TP and FP rates using two features (μ , σ) of sensors S_1 and S_2 with an observation period of 2 and 0.25 seconds. This figure further highlight that there was confusion between Position 3 and Position 4. In fact, 7% Position 4 (right lateral) was misclassified as Position 3 (supine), when the observation period was fixed to 0.25 seconds. On the other hand, Positions 1, 2, and 3 were well recognized (100% of matching rate).

5. Conclusions

In this work, a service-oriented middleware platform for AAL for green bed activity monitoring has been presented. The key idea of this work is to leverage the presence of a WSN by collecting the RSS measured among fixed general-purpose wireless devices, deployed in the environment, and a wearable one. The RSS measurements are used to classify a set of user's positions in the bed, monitoring the activities of the user, and thus supporting the bedsores and the sleep monitoring issues.

In particular, measurements showed that it is possible to use low-cost transceivers to classify the patient's positions. Good classification performance can be achieved by using only the received signal strength measurements relevant to a wearable and a fixed sensor. In fact, by using the RSS traces measured by only one sensor (S_1) at 8 Hz with an observation

period of 2 seconds, 100% matching rates were achieved even with only one feature (μ). These results are similar to those showed in [7], but when the energy consumption is taken into account the proposed solution increases the performance with respect to the state of the art. In fact, measurements showed that it is possible to achieve 100% matching rate at 1 Hz, increasing the mobile lifetime of about 85% with respect to [7]. Instead, when the requirement of responsiveness increases (i.e., observation period equal to 0.25 seconds), the bed position detection service needs to exploit the RSS traces measured by at least two sensors. Our analysis suggests that, in the near future, the electrical sockets in the environment can be also exploited from bed activity monitoring services. In particular, in our experiments two sensors near to the bed on the night table and one sensor on the dresser are able to provide the user's positions in the bed. The LOS condition guarantees the 100% of matching rate when the fixed nodes are placed close to the user (as on the night table).

Conflict of Interests

The authors do not have any conflict of interests with the Weka program exploited in this paper to evaluate the performances of the proposed system.

Acknowledgments

This work was supported in part by part by the European Commission in the framework of the universAAL FP7 project (Contract no. 247950) and GiraffPlus FP7 project (Contract no. 288173).

References

- [1] M. Tentori and J. Favela, "Activity-aware computing for health-care," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 51–57, 2008.
- [2] A. A. P. Wai, K. Yuan-Wei, F. S. Fook, M. Jayachandran, J. Biswas, and J.-J. Cabibihan, "Sleeping patterns observation for bedsores and bed-side falls prevention," in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine (EMBC '09)*, pp. 6087–6090, September 2009.
- [3] J. Baek, G. Lee, W. Park, and B.-J. Yun, "Accelerometer signal processing for user activity detection," *Knowledge-Based Intelligent Information and Engineering Systems*, Springer, vol. 3215, pp. 573–580, 2004.
- [4] A. M. Adami, M. Pavel, T. L. Hayes, and C. M. Singer, "Detection of movement in bed using unobtrusive load cell sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 481–490, 2010.
- [5] H.-Y. Lau, K.-Y. Tong, and H. Zhu, "Support vector machine for classification of walking conditions using miniature kinematic sensors," *Medical and Biological Engineering and Computing*, vol. 46, no. 6, pp. 563–573, 2008.
- [6] A. R. Guraliuc, P. Barsocchi, F. Potorti, and P. Nepa, "Limb movements classification using wearable wireless transceivers," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 474–480, 2011.
- [7] P. Barsocchi, "Position recognition to support bedsores prevention," *IEEE Transactions on Information Technology in Biomedicine*, vol. 99, p. 1, 2012.
- [8] M. Quwaider and S. Biswas, "Body posture identification using hidden markov model with a wearable sensor network," in *Proceedings of the ICST 3rd International Conference on Body Area Networks (BodyNets '08)*, pp. 19:1–19:8, 2008.
- [9] P. A. Bernstein, "Middleware: a model for distributed system services," *Communications of the ACM*, vol. 39, no. 2, pp. 86–98, 20131996.
- [10] S. Krakowiak, *Middleware Architecture with Patterns and Frameworks*, INRIA, Rhône-Alpes, France, 2007.
- [11] E. Parry and T. Strickett, "The pressure is on—everyone, everywhere, everyday," in *Workshop on ARATA*, pp. 589–592, June 2004.
- [12] "National sleep foundation," <http://www.sleepfoundation.org/>.
- [13] A. Oksenberg and D. S. Silverberg, "The effect of body posture on sleep-related breathing disorders: facts and therapeutic implications," *Sleep Medicine Reviews*, vol. 2, no. 3, pp. 139–162, 1998.
- [14] E. Hoque, R. F. Dickerson, and J. A. Stankovic, "Monitoring body positions and movements during sleep using wisps," in *Wireless Health 2010*, pp. 44–53, ACM, New York, NY, USA, 2010.
- [15] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, p. 21, 2012.
- [16] M.-R. Tazari, F. Furfari, Á. Fides-Valero et al., "The universal reference model for aal," in *Handbook of Ambient Assisted Living: Technology For Healthcare, Rehabilitation and Well-Being*, vol. 11 of *Ambient Intelligence and Smart Environments*, pp. 610–625, IOS Press, 2012.
- [17] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S - A publish/subscribe protocol for wireless sensor networks," in *Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COM-SWARE '08)*, pp. 791–798, January 2008.
- [18] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, 2002.
- [19] "Javascript object notation," <http://www.json.org/>.
- [20] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [21] "Tunstall healthcare," <http://www.tunstall.co.uk/>.
- [22] "Cleode embedded systems," <http://www.cleode.fr/>.
- [23] "Crossbow technology inc," <http://www.xbow.com/>.
- [24] F. Potorti, A. Corucci, P. Nepa, F. Furfari, P. Barsocchi, and A. Buffi, "Accuracy limits of in-room localisation using RSSI," in *Proceedings of IEEE International Symposium on Antennas and Propagation (APSURSI '09)*, pp. 1–4, June 2009.
- [25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [26] S. R. Gunn, "Support vector machines for classification and regression," Tech. Rep. 256459, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, May 1998.
- [27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

