

RESEARCH

Open Access



A deep learning approach for pressure ulcer prevention using wearable computing

Giovanni Cicceri^{1*} , Fabrizio De Vita¹ , Dario Bruneo¹ , Giovanni Merlino¹  and Antonio Puliafito^{1,2} 

*Correspondence:
gicceri@unime.it

¹ Department of Engineering,
University of Messina,
Messina, Italy
Full list of author information
is available at the end of the
article

Abstract

In recent years, statistics have confirmed that the number of elderly people is increasing. Aging always has a strong impact on the health of a human being; from a biological point of view, this process usually leads to several types of diseases mainly due to the impairment of the organism. In such a context, healthcare plays an important role in the healing process, trying to address these problems. One of the consequences of aging is the formation of pressure ulcers (PUs), which have a negative impact on the life quality of patients in the hospital, not only from a healthiness perspective but also psychologically. In this sense, e-health proposes several approaches to deal with this problem, however, these are not always very accurate and capable to *prevent* issues of this kind efficiently. Moreover, the proposed solutions are usually expensive and invasive. In this paper we were able to collect data coming from inertial sensors with the aim, in line with the Human-centric Computing (HC) paradigm, to design and implement a non-invasive system of wearable sensors for the prevention of PUs through deep learning techniques. In particular, using inertial sensors we are able to estimate the positions of the patients, and send an alert signal when he/she remains in the same position for too long a period of time. To train our system we built a dataset by monitoring the positions of a set of patients during their period of hospitalization, and we show here the results, demonstrating the feasibility of this technique and the level of accuracy we were able to reach, comparing our model with other popular machine learning approaches.

Keywords: Wearable technology, AI, Machine learning, Deep learning, Healthcare system, IoT, Keras, TensorFlow, Cloud

Introduction

The aging of the human population has a profound impact on all aspects of social life, with particular reference to healthcare processes. This fact involves a variety of preventive actions aimed at delaying and contrasting decline in the daily activities of each individual from a physiological point of view. One of the biggest challenges that healthcare professionals try to face is the prevention of pressure ulcers, which are one of the most periodic and feared consequences of aging.

Pressure ulcers (PUs), also called *bedsores* or *pressure sores*, are injuries whose formation happens when the skin is under constant pressure for a long enough period of time, and it occurs more frequently in hospitalized patients. Data from the National Advisory

Panel (NPUAP), a multidisciplinary group of experts in pressure injury, shows that PUs occur with a frequency between 10 and 18% in intensive care units, between 2.3 and 28% in long-term care units and between 0 and 29% in home care units [17]. PUs, which usually cause pain, reduce patients mobility, and worsen their quality of life, appear as ischemic lesions on the skin and underlying tissues, caused by the prolonged pressure between two hard surfaces. Depending on the extent of the lesion, PUs are categorized into different grades [11] of severity.

In the first grade, the skin presents a pink or red color that disappears in 30 s after removing the pressure while in the second grade the skin loses continuity in correspondence with the appearance of vesicles. With respect to the third grade, the skin presents a severe decline in the state of the tissue and in the fourth grade, there is a total loss of skin density together with a muscle injury. Depending on the level of loss, it is sometimes necessary to remove the infected tissue to prevent sepsis from developing and contaminating other areas of the body.

The total risk for each person to develop PUs can be estimated in accordance with different scales: one of these is the Braden scale, which is based on several risk factors, such as poor motion activity, but also temperature, skin humidity and malnutrition [14]. PUs negatively affect the quality of life of patients and prolong their recovery process thus increasing the costs for the healthcare systems. It is therefore necessary to implement and adopt preventive strategies to solve this problem steadily getting more common.

There are several systems on the market that allow the monitoring of risk factors related to PUs, but until now no suitable solutions have been proposed, especially for people with a medium risk of PUs. Healthcare techniques [29] aimed at personalized patient treatment introduce a series of new solutions that act as a bridge between patients and medical services. A possible approach, for solutions of this kind, comes from wearable technologies, and the choice of using wearable devices is actually a significant step towards improving patient health, also considering that these devices are already available in the market and affect our lives under several aspects.

Wearable technology introduces a new paradigm in digital communications: unlike other portable devices, such as smartphones and tablets, wearable devices are mostly not supposed to be interacted with in order to generate and collect data. This radically changes the way we approach surrounding environments, deal with interpersonal relationships, and information we exchange with other people, by extending the reach of what is called the Internet of Things (IoT) [13]. IoT represents a key factor used to acquire information in order to provide the best medical assistance to patients [15]. The community refers to this topic through the term Internet of Medical Things (IoMT). IoMT is one of the main drivers that the field of medicine is going to embrace in the near future. It represents an important enabler for e-health in clinical decision support systems, and it is fundamental to develop intelligent systems, meant to improve processes for patient care by generating significant clinical data [9].

In such a context, *cloud computing* and *machine learning* technologies play an important role and can be used in order to manage complex systems with a large amount of data. The former is fundamental for the management of data collected by IoMT devices that use a different kind of sensors, providing also the computational power to process it. On the other hand, *machine learning* represents a key approach to build a model that

is able to correlate all data from different devices and to respond dynamically to a wide range of different situations, without the help of human beings.

To provide a more accurate and simplified medical monitoring, this article proposes an automatic deep learning approach at the basis of a system which is able to recognize motor activities of patients and send real-time activity predictions by helping the medical staff to assess the risk of PUs formation on a specific patient. The use of machine learning as a paradigm for the design of a medical support system is starting to be considered a valid approach for this kind of solutions, especially when the data to analyze is hard to correlate. A survey about machine learning, and in particular deep learning techniques applied to the domain of biomedical disciplines is available at [5].

The proposed system collects information from various wearable sensors incorporated and worn by the patient, it analyzes the data to create, and periodically visualize, the risk assessment for PUs. Thanks to this, the medical staff can easily see the status of each patient in real time, by allowing the former to provide better assistance.

The contribution of this paper is twofold: (i) we realized a labeled dataset containing the readings coming from a set of inertial sensors (accelerometer and magnetometer) together with the real time patient position associated, which has been necessary for the design and the training of the model; (ii) we designed a system that allows us to predict the user position over time with a high level of accuracy and with a minimal hardware usage when compared to other solutions found in the literature. Moreover, our goal was also to build a system which is totally transparent to the patient, and for this reason, we used a specially-crafted *hospital gown*, containing a set of sensors that do not require any kind of invasive attachment to the body of the patient, giving her/him the possibility to move freely without any constraint.

This paper is organized as follows: "[Related work](#)" section describes the state of the art and related works about machine learning techniques and wearable technologies applied to healthcare, and specifically to PU-related problems and their applications; "[System architecture](#)" section describes the overall architecture of each part of the system; "[Data management](#)" section describes its data management; "[Experimental results](#)" section presents an experimental overview, including the results obtained, both in terms of accuracy for the machine learning model, and of the development of the web interface, to allow users to visualize some information about patients, and the corresponding risk of developing PUs; "[Conclusions](#)" section closes the paper and gives some details about future works.

Related work

In recent years we have witnessed ever higher development of research in pervasive systems applied to healthcare, that employ smart devices to support the daily activities of patients, through pervasive computational applications, according to the needs of users, which also take into account changes to the environment. Current convergent research on pervasive systems is making a major effort on the recognition of patient activities, which is a very important aspect in ensuring satisfactory healthcare, thus enabling medical personnel to provide appropriate care to patients. In addition to this statement and in accordance with [7] the technology employed as a primary choice to capture human activities serves not only to place people at the center of activities related

to new technologies but also to study their behavior and interaction with others using approaches based on monitoring of physiological and behavioural signals that contribute to a better understanding of human activities especially in elderly patients and in-home care settings [24].

In such context, wearable technology is increasingly being integrated into the ecosystem of e-health solutions with scope into the overall well-being of patients, and specifically into prevention and treatment of chronic patient diseases. The use of this technology allows personalized healthcare through the sharing of distributed information, by favoring better prevention, and real-time monitoring of vitals and behavioral data related to everyday life. A high number of healthcare applications are expected to take advantage of wearable technology over the next two decades. Authors in [16] propose a system of recognition of human activities based on wearable sensors by using a two-stage end-to-end convolutional neural network and a data augmentation technique. Another typical recent example in [2] is the continuous glucose monitoring that allows diabetic patients to monitor blood glucose levels by learning from measurements on their body functions helping the patient to self-manage and to proceed with the treatment independently. For this same problem, the authors in [25] introduce a clinical approach tested on nine types 2 diabetic patients through the use of wearable devices that use optical signals to obtain a photoplethysmogram (PPG) adopted to estimate glucose values by building Random Forest and Adaboost regression models obtaining 90% of accuracy.

The use of technology for the prevention of bedsores is one of the key issues facing healthcare to limit the costs of this problem. An example comes from [22], where the authors developed a wearables wireless patches system by placing them directly in risk areas on the skin of the subjects by monitoring contact pressure, temperature, humidity and movement and sending the values to a base location. Authors in [6] developed a multimodal sensor system for the evaluation and treatment of PUs that can provide spectral, thermal and chemical analysis, the real-time vision of injury size and prediction of wound healing in support of hospitals, clinics and home environments. Authors in [20] propose a technique for the prevention of PUs that helps patients effortlessly and independently, which comes from a system that provides electrical stimulation to the pressurized points of the gluteal muscles through a smartphone application connected wireless to the head and regulates the electrotherapy parameters. For medical analysis, the therapy history track is stored in the cloud. Another example is provided in [10], where a wearable wireless monitor continuously measures the duration and orientation of the posture of the patient through an accelerometer built into the torso and adaptive posture algorithms accurately measure the angle of the spine vertebral in order to classify the orientation of the patient. Other approaches in [14] for the PUs prevention focus more on detecting long-term pressure to particular skin regions of the body. All these systems use pressure sensors because pressure is one of the main important factors for the development of PUs. Authors in [26] present a smart remote monitoring system developed through a ZigBee network infrastructure with pressure sensors located on the bed for the prevention of PUs in patients with reduced mobility.

A work to support pressure sores issue was introduced in [3] in which low-cost 2.4 GHz wireless transceivers were used to recognize the different activities. The idea has

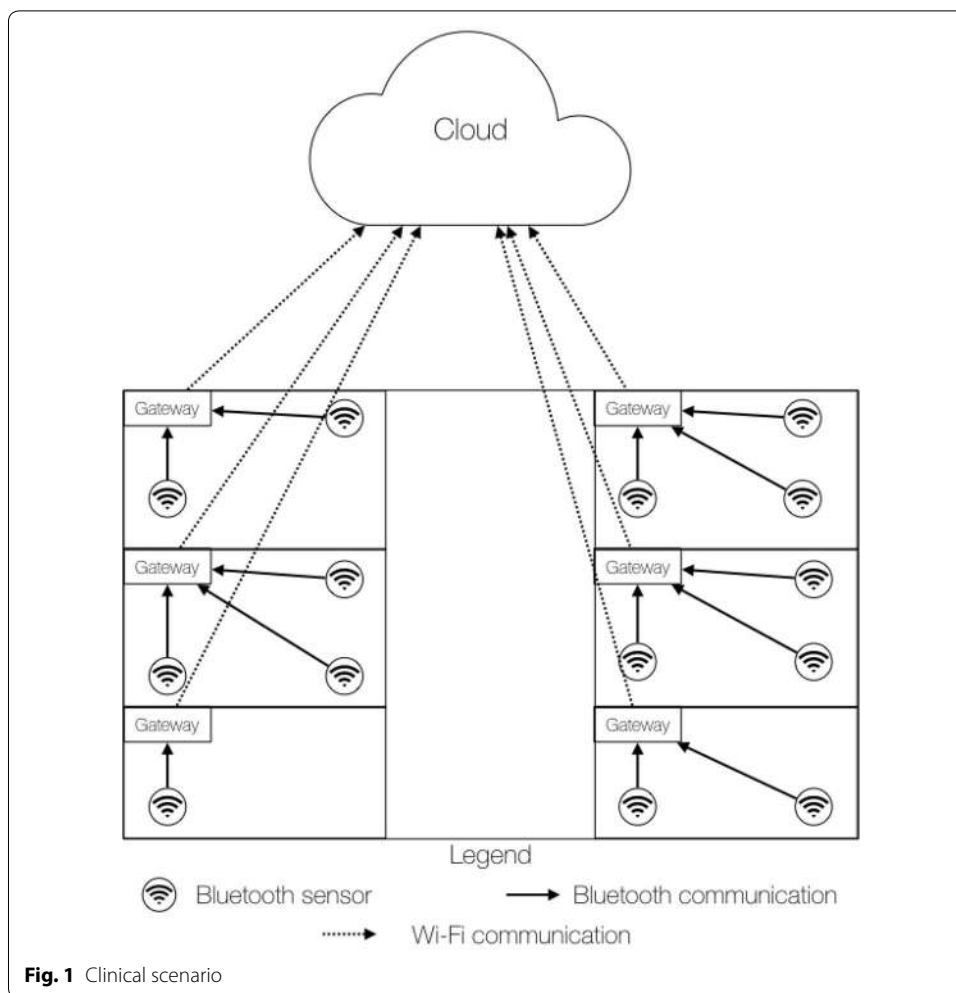
been to collect the intensity of the received signal (RSS) measured between the fixed and wearable devices and to use Support Vector Machines (SVMs) and K-Nearest Neighbor (KNN) methods to classify the different positions. Another work that enables healthcare professionals to provide better patient care in less time was the development of a software–hardware platform [28] that collects sensor data embedded in the bed, analyzes this information and builds a pressure distribution map on all over the body with periodical indication and controls the bed actuators to adjust the profile of the bed surface to redistribute the pressure on the whole body. This platform uses SVMs algorithm as a classifier to determine the level of risk of developing ulcer.

There are other recent approaches that have attempted to create an efficient monitoring of people's health activity. Authors in [21] proposed an innovative multi-sensor fusion system that improves the performance of human activity detection using a multi-view ensemble method to integrate the expected values of different motion sensors using different classification algorithms, such as logistic regression, K-Nearest Neighbors (KNN) and Decision Tree (DT). Authors in [23] use the features of two different radar systems operating at C band and K band through a Support Vector Machine classifier to recognize 10 human activities for remote health monitoring purposes. Another research [19], consisting in the recognition of human activities through video sequences captured by UAVs, uses two different approaches: (1) an offline phase in which the human/non-human model and a human activities model are built through a convolutional neural network, and (2) an inference phase that uses these models to detect human beings and recognize their activities. Another study of activity recognition is proposed in [30]: here authors presented a system which uses a multi-sensor network positioned in different body parts, composed by 9 inertial measurements, from accelerometer, magnetometer, gyroscope, and quaternion sensors, respectively, by applying feature extraction in the time and frequency domain, and two classification algorithms, namely a DT and a Random Forest (RF), achieving a 99.1% of accuracy.

All these systems we presented do not meet the basic requirements we captured and have some disadvantages, including the high cost and the way of positioning the sensors in the bed making vulnerable the comfort of the patient. In particular, our intent is to build a non-invasive system which it has to be easy to use, accurate, and with a minimal hardware usage in order to reduce the costs. For this reason, instead of using pressure sensors and exerting pressure directly on areas of the skin or on the bed, we thought to monitor the patient with wearable sensors with the support of machine learning techniques in order to estimate his/her position over time and assess the risk of PUs formation.

System architecture

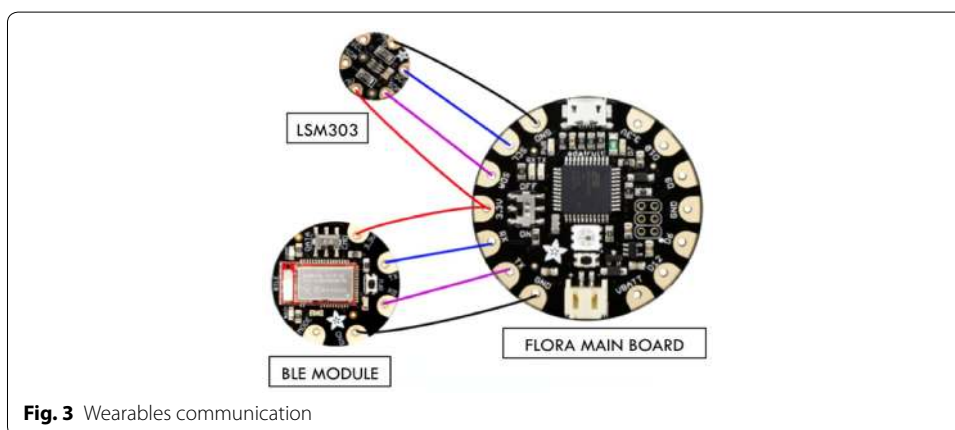
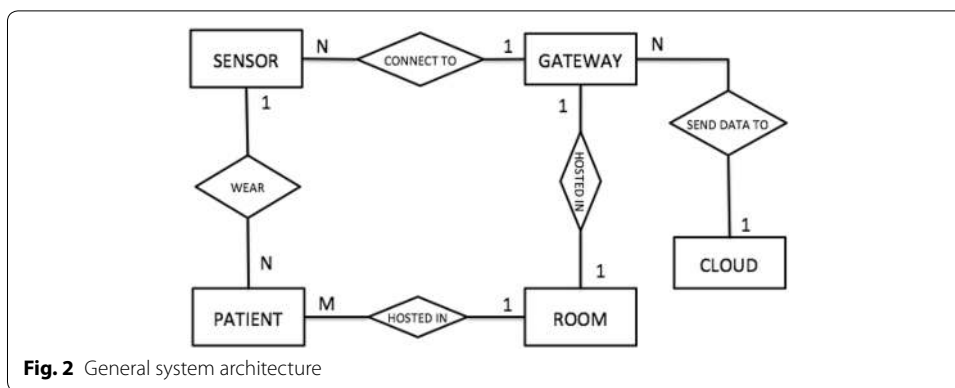
In this section, we introduce the scenario in which we operated in order to build a decision support system using IoMT to monitor and evaluate the risk of PUs formation. Our goal is to build a non-invasive system that can be used by anyone in a very simple way. Figure 1 shows the clinical scenario we imagined, which consists of one or more patients located in a room wearing a hospital gown with attached a set of wearable sensors in the abdominal area. The attached sensors are a three-axis accelerometer, a three-axis magnetometer that we use in order to detect the patient motion and measure postural body



orientation. In order to manage several wearable devices belonging to different patients in the same room, we propose a hardware/software architecture that is composed by a room gateway that behaves as a bridge among the IoMT devices and the cloud.

The architecture we propose has been chosen in order to improve the scalability of the project: Figure 2 shows how the system architecture is organized. The main function is performed by the gateway, which has to collect data from the wearable sensors attached to the patients bodies, process it and then send the information to the remote server which is used to read, interpret, and store it in a database which can be accessed by the medical staff to visualize the risk of PUs formation of each patient. In this sense, deep learning has been used in order to build a model that takes as input the raw data and generates as output the more probable activity or position associated to it.

We can see that the relationship between the gateway and wearable devices is “1-to-N”, which means that the gateway may get data from several wearable sensors. This is a realistic assumption because in the clinical scenario we have in mind, different sensors could be worn by different patients. With respect to the relationship between gateway and the cloud an “N-to-1” relationship has been envisioned: this means that each gateway sends data to one remote server only hosted in the cloud, and any server could get data from

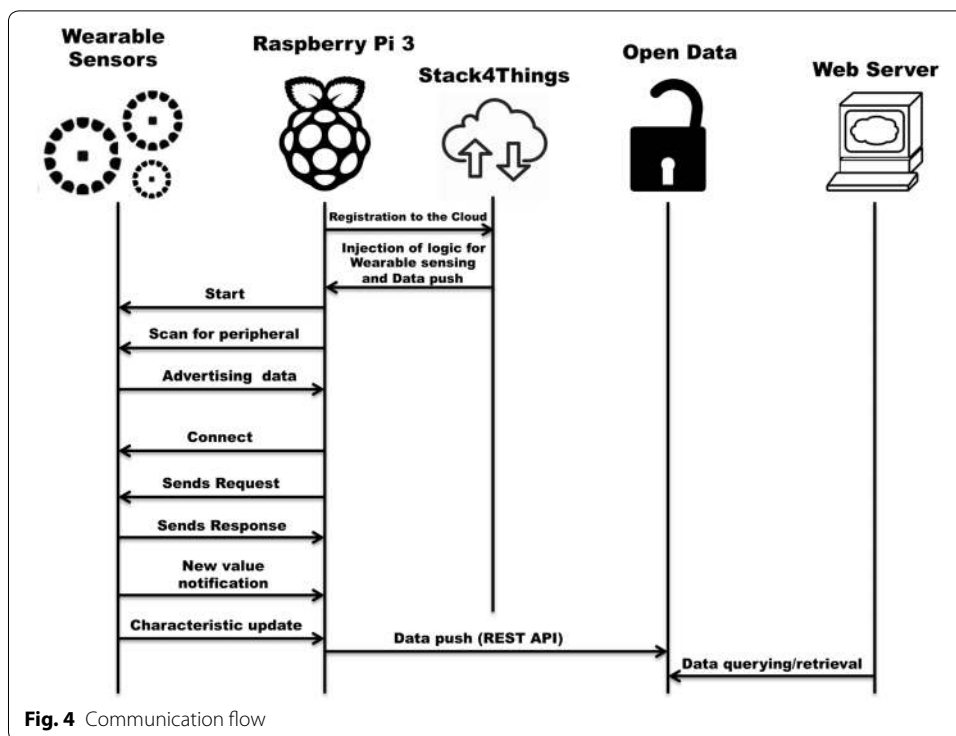


several gateways. Also, this may be considered an accurate hypothesis for the system under consideration, because one gateway should be placed in each room and the single remote server should be accessible to authorized users. Finally, with respect to the relationship between the patient and the room, it is a “M-to-1” relationship meaning that one or more patients can be hosted in a single room.

System implementation

In this subsection, we present the implementation of the above described architecture. With respect to the room gateway, we implemented it using a Raspberry Pi 3, which is a small platform that supports a complete Linux system with reduced costs. Figure 3 depicts the wearable devices that perform another main function of the system: in particular, we used the Flora device by Adafruit, an electronic platform compatible with Arduino combined with the LSM303 integrated circuit which contains an accelerometer and a magnetometer. Their job is to send raw accelerometer and magnetometer data values to the Raspberry Pi through the Bluetooth Low Energy (BLE) module.

The communication flow between the three components is regulated by Generic Access Profile (GAP) and Generic Attribute Profile (GATT) procedures. GAP represents the generic procedure for discovering BLE devices by checking connections and advertising, which determines the way in which two devices can or not interact with each



other. GATT defines the mode and procedures in which two BLE devices transmit data using the attribute protocol (ATT): wearable devices act as a GATT server and Raspberry as a GATT client. GATT transactions in BLE are based on high-level and nested objects named profiles, services and characteristics. Profiles consist of services that cluster raw data (characteristics); in turn, a characteristic encapsulates a single data point and consists in two attributes (declaration and value). Attributes are the primary data described in the ATT protocol [27].

In terms of Cloud, we adopted as solution Stack4Things (S4T), which is an OpenStack-based framework for Infrastructure-as-a-Service Cloud computing. Figure 4 shows the communication flow between the wearable devices, the Raspberry, the S4T Cloud, an Open Data CMS (Content Management System) and a web server. A Raspberry, after the registration to the Cloud and after receiving logic for wearable sensing by Cloud-powered *injection*, starts the communication process scanning the surrounding for BLE wearable devices that indicate their presence through advertising data. Once the Raspberry finds the device of interest, it connects to it using the BLE communication. As soon as the Raspberry connects to wearable device, it will stop to send advertise data and will start to send raw data in real time according to the position of patients. In parallel, and in real time, Raspberry processes, analyzes, and pushes the data to an Open Data datastore, provided by the CKAN CMS, using its REST API.

On the Cloud side, the training of our deep learning model is performed offline (a detailed description will be given in "Data management" section). At the beginning, the system collects the data through the Open Data datastore in order to create a labeled dataset, then used to train the deep learning model. Once we obtain the model, it is deployed in the Raspberries, that perform the real-time inference process by feeding

the pre-trained model with the data gathered from the sensors via BLE communication. Once the data has been transferred to the Open Data service, the web server will be able to retrieve the data for visualization and to be analyzed by the medical staff. These processes continue until the medical staff will cut off transmission with a specific patient by stopping the communication between Raspberry and the wearable devices.

Providing some context about our choice for the Cloud, S4T [18] is an IoT/Cloud integration framework which originates as research effort in IaaS-oriented IoT *cloudification*. The bottom-up, implementation-driven approach to this project targets a very successful, open source, middleware for IaaS Clouds, namely OpenStack, in order to fully explore the I/O cloud [4] paradigm. Single-Board Computers (SBC) such as Raspberry are perfectly suited to host Python-based *lightning-rod* (LR), which acts as a board-side agent for S4T, interacting with its cloud-side counterpart, IoTronic, through WebSockets-powered full-duplex communication.

Leveraging the I/O cloud approach, IoT infrastructure administrators are able to inject logic, in the form of LR *plugins*, into IoT nodes at runtime. Plugins are useful to manage and expose IoT-hosted resources, such as on-board sensors, or BLE-connected ones, as a service from the Cloud, regardless the configuration or level of connectivity of nodes hosting them. In particular, LR implements a plugin loader, fully managing the lifecycle of plugins: business logic injected from the Cloud is received, validated and run by the plugin loader in order to implement specific user-defined actions. New REST resources are automatically created exposing user-defined commands on the Cloud side: as soon as such resources are invoked, the corresponding code is executed board-side. In our case study, two plugins are injected, which carry the logic to, respectively, drive the BLE interactions, and push any newly received samples from sensors to an endpoint of choice, i.e., our CKAN instance.

Data management

In this section, we provide more details about the dataset and the design of the deep learning algorithm: the idea to adopt a deep learning approach comes from the fact that in such a context can be very difficult to detect the patient position accurately. In fact, problems related to recognition of patient motion activity highlights some issues related to the high variability in the motion patterns. Think, for example, to the movements of an older adult and a younger one: even if they move seemingly in the same way, the actual execution of motion, i.e., its fine-grained sequence, can be very different. Another big issue is related to the sensors, which are attached to the hospital gown irregular surface: in particular, the impossibility to fix in position the sensors can sometimes lead to the generation of dirty and wrong data, which requires a huge and time consuming preprocessing phase before the use. For all these reasons, we decided to face this classification problem by using a deep learning approach to develop a system which is capable of understanding the complex relationship between the data coming from the sensors and the user motion. The key idea behind deep learning consists of connecting in cascade several hidden layers between the input and the output. By doing so, this technique is able to discover very intricate relationships between the input and the output thanks also to the features extraction process which allows a Deep Neural Network (DNN) to



get rid of all those features which are redundant or not informative for the training process thus reducing the model complexity and training time [12].

In such a context, we worked on a supervised approach, where all data stored inside the dataset are labelled with the corresponding patient position. Data collection has been performed by equipping six volunteers, belonging to different profiles—in terms of age, weight, and sex—with a hospital gown. The latter had a set of wearable devices attached, recording the body position while performing five physical activities such as: staying seated, prone, supine, laying on the right and left, and in motion. For privacy reasons, we have avoided inserting the real names and surnames of the patients, also avoiding the appearance of their faces in photos.¹

The sensors were placed on the chest of each patient (as shown in Fig. 5 for some patients) and, thanks to the help of nurses who, through a camera, monitored all patients, the activities were collected in medicine department with no constraints on the way these must be executed. Every volunteer had no constraints on the movements that he/she could do, so the data collected derives from a totally natural motion scheme. This, in particular, was the main requirement we asked for to the volunteers in order to obtain a dataset without any kind of bias. Moreover, we decided to collect data from different persons in order to have a better generalization of the common motion patterns.

So far, we discussed how we collected the data, let us now introduce a formal definition for a single record contained inside the dataset. From a mathematical point of view, it can be considered as a tuple composed by: the patient id, the timestamp, the sensor data collected from the accelerometer and the magnetometer, and the label representing the user position at that timestamp. It can be expressed as follows:

¹ Involved patients who submitted to tests undersigned a document for the clearance with regards to processing of their personal data.

$$d_t = \langle (Id, Time, Ax, Ay, Az, Mx, My, Mz, Pos) \rangle \quad (1)$$

where d_t represents the datapoint recorded at timestamp t . Since our goal is to understand the patient position to prevent the generation of PUs while he/she is resting in the same position for too long a period of time, we can reduce the number of positions in which we are interested. By analyzing the PU state of the art together with the help of specialized personnel, we were able to localize the main body areas where the ulcers usually appear, and correspondingly the hazardous positions. At the end of this analysis, we defined a set \mathcal{P} containing the positions that we have to monitor:

$$\mathcal{P} = \langle Supine, Prone, Right, Left, Sitting, Movement \rangle. \quad (2)$$

The final dataset \mathcal{D} is obtained by merging all the datapoints d_t at different timestamps (see Table 1), where each row contains all the data registered by the sensors attached to the patient body, together with the corresponding position label. At the end of the procedure, we collected a total of 8708 samples, which is a fairly good amount of data to train our model.

Then, we split the “original” dataset into three sets, namely: *train set*, *validation set*, and *test set*, where the train and test set dimensions are, respectively, 85% and 15% of the original dataset we collected, while the validation set has been obtained by taking 15% of the training data.

From an architectural point of view, a DNN can be considered as a traditional artificial neural network (ANN) with a large number of hidden layers between the input layer and the output one. These architectures allow DNNs to fit very complex non-linear relationships. However, at the same time, they make more difficult the training process since they require a large amount of training data in general because these models tend to overfit the data. Hence, it is necessary to adopt the right countermeasures to prevent this behaviour in order to obtain a good model.

Figure 6 depicts the DNN architecture we designed, where the input layer consists of the measurements gathered from the sensors. In contrast, the output one consists of the six different positions (i.e., Supine, Prone, Right, Left, Sitting, Movement) we want to classify. It is worth mentioning that each DNN neuron (except for the input layer) is also provided with bias values (not shown in the Figure). The latter are mainly used to give more “elasticity” to the network in terms of predictive capability, but also to avoid the presence of “dead neurons” whose output is 0. After trying several models, we found out that four hidden layers are enough to represent the relationship between the input and the output fully. In general, all the hyper-parameters have been tuned empirically by doing multiple tests with the support also of a validation set. With respect to the number of hidden units, we started with an architecture with 20 neurons and decreased the number of the neurons by five units thus converging to the output layer size which is equal to 6. The idea to start with a number of neurons, which is larger than the input size and decreases as long as we go deep, is a widespread choice adopted in deep learning circles.

The learning rate value is a standard choice since it allows us to reach the global minimum with a “reasonable” number of training epochs. Regarding the activation function, we used the Rectified Linear Unit (ReLU) which in the recent period

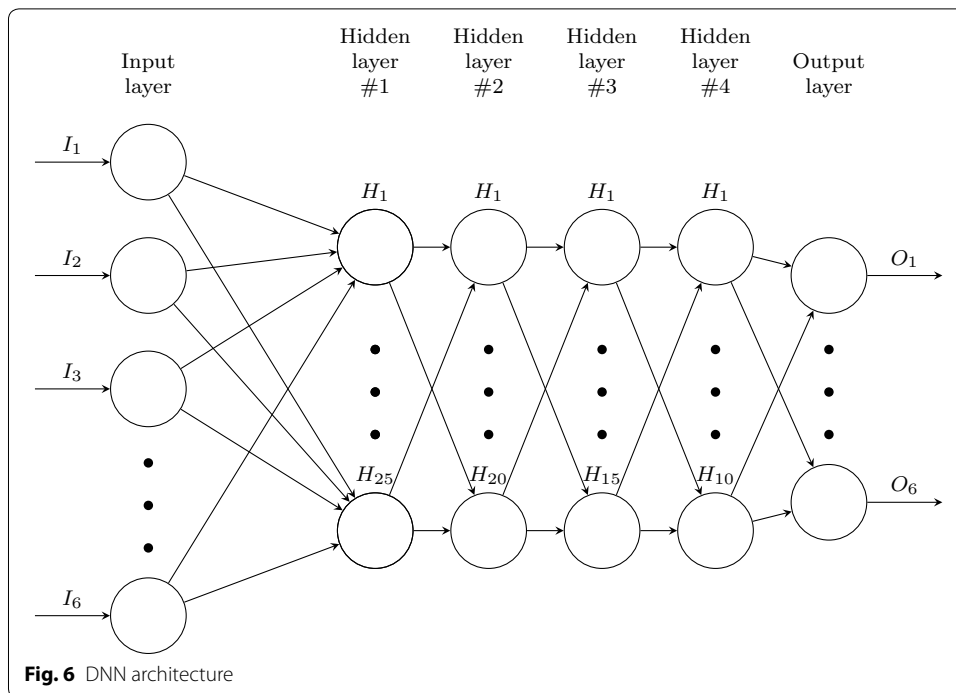


Table 1 Labeled patient position dataset structure

Id	Timestamp	Ax	Ay	Az	Mx	My	Mz	Pos
0	2018/07/22 19 : 37 : 15	-0.04	0.24	9.77	26.82	8.18	-1.02	Supine
1	2018/07/24 19 : 42 : 15	7.53	-5.41	3.14	-14.73	40.27	8.98	Left
...
4	2018/07/27 22 : 37 : 15	-4.67	-8.87	1.80	4.73	21.91	-33.16	Sitting
5	2018/07/30 23 : 23 : 45	8.43	4.98	0.51	11.45	-35.36	3.67	Right

is becoming the most used one basically because it speeds up the training process requiring a lower number of training iterations if compared with other activation functions like the Sigmoid, and at the same time, it allows to solve the *vanishing gradient* problem which is typical for those activation functions that saturate for certain inputs. Detecting the patient position given a set of sensors measurements can be categorized as a supervised multi-class classification problem. As already mentioned above, the input provided to the network consists of the set of measurements gathered by sensors. Such data is then passed to the hidden neurons that perform the feature extraction process through Eq. (3):

$$y = \sigma \left(\sum_{i=1}^n x_i \cdot w_i + b \right) \tag{3}$$

where σ is the activation function, x_i is the neuron input, w_i is the weight value of the connection between the input x_i and the neuron itself, and b is the bias of the neuron, a constant value used to avoid the output y to be 0.

Table 2 Deep neural network parameters

DNN parameters	
Number of hidden layers	4
Number of neurons	[25, 20, 15, 10]
Input dimension	6
Output dimension	6
Learning rate	0.001
Training epochs	500
Regularization parameter	0.01
Activation function	ReLU

Table 3 Experimental environment and specifications

Experimental environment	
Processor	Intel Core i7 2, 9 GHz
Operating system	OS X Mojave
Developing environment	Atom 1.38.2 x64
Programming language	Python 3.65
Model package	Keras 2.3.1
Backend	Tensorflow 2.0

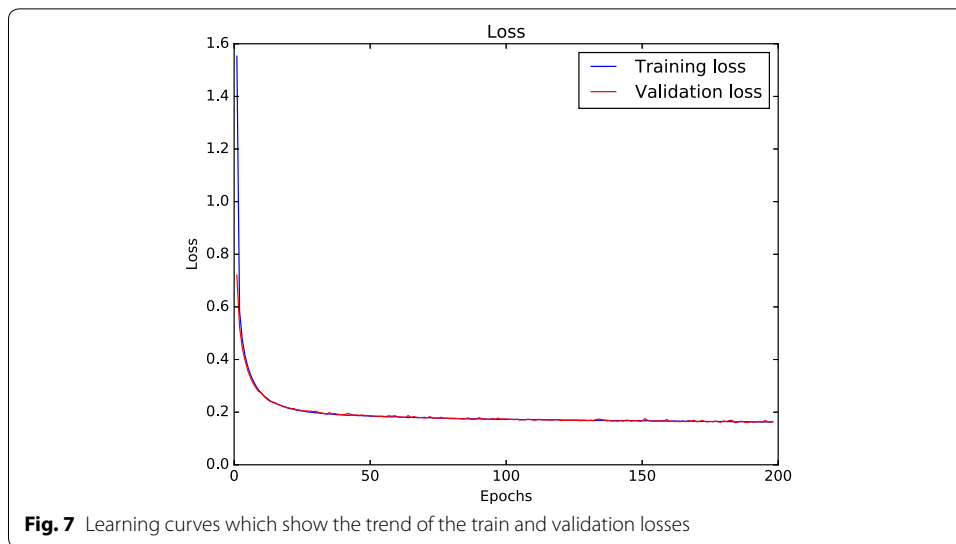
In the last layer of the proposed DNN architecture, we adopted Eq. (4) which represents the *softmax* activation function whose task is to “compress” the output of each neuron in the output layer so that the sum of these values adds up to 1. Here the \tilde{y}_j represents the value of the *j*th output neuron that can be considered as the probability of the corresponding class given a specific input. In this sense, to obtain the position of the patient, the DNN applies the *argmax* function to the output layer in order to extract the most probable class associated with the input fed into the network.

$$\tilde{y}_j = \text{Softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^p e^{z_k}}, \quad j = 1, \dots, p. \quad (4)$$

Finally, we used a technique called regularization, which consists in applying a penalty term to each weight of the network in order to prevent it from overfitting. In general, it is necessary to define a regularization term which determines the penalty we want to assign to each term. From the test we performed, we concluded that a value equal to 0.01 provides a suitable penalization level, preventing the overfitting of the network. Moreover, we adopted an early stopping process to prevent overtraining, that in some cases could lead the network to overfit as a collateral effect. Table 2 shows the parameters used for the network we designed.

Experimental results

In this section, we present the obtained results by making a set of 100 experiments to test the performance in terms of accuracy, training, and inference times. In Table 3, we present the experimental environment settings in which we operated and the tools we



used. To test the architecture of the DNN explained in "[Data management](#)" section, we prepared the following test environment: we trained the DNN on a MacBook Pro with i7 Processor under the MacOS operating system (Mojave release), as development environment we used Atom, and Python as programming language.

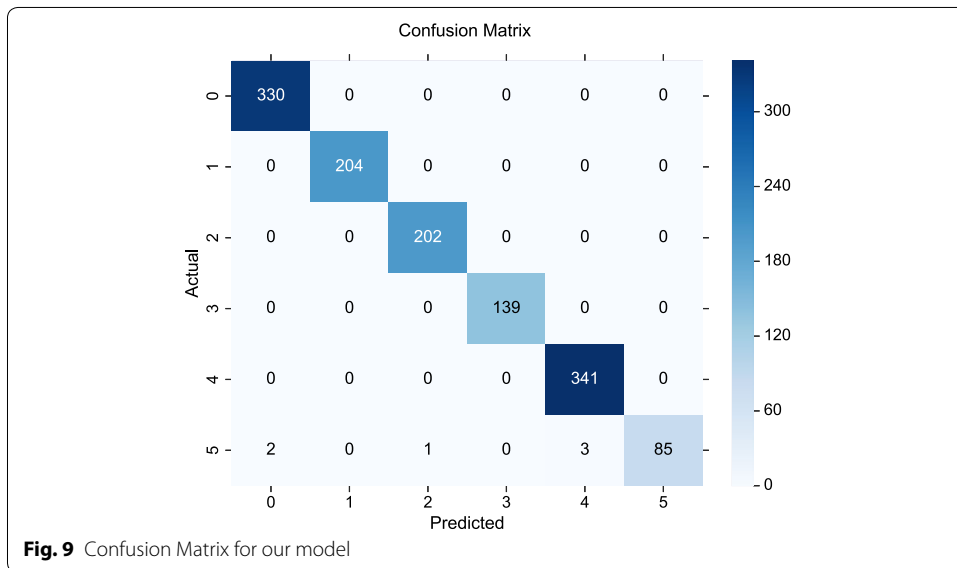
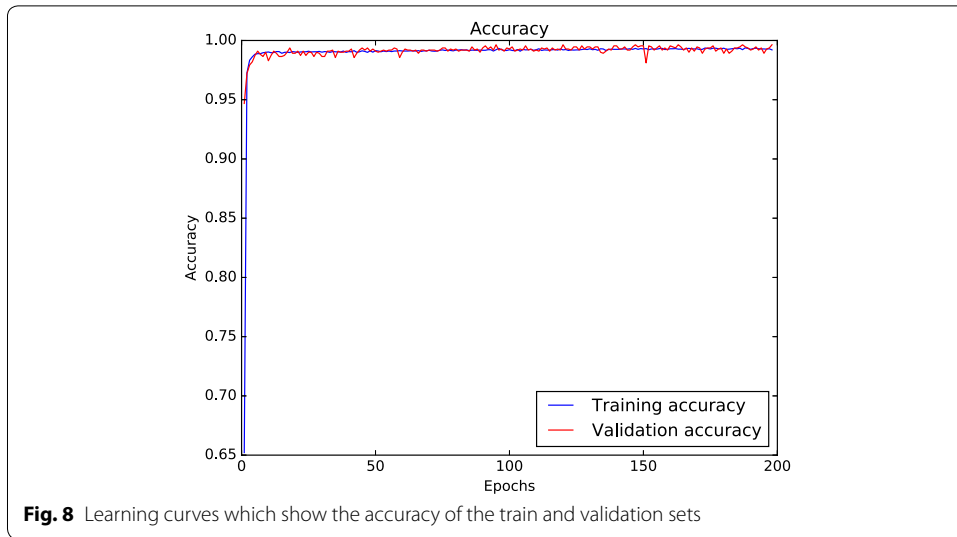
Once all the data has been collected, we used Keras [8] to train our model and generate the position predictions. In particular, Keras is a high-level API written in Python, which runs on top of several machine learning frameworks like TensorFlow [1]. During the learning phase, wearable sensors send the raw values with the label indicating the position and the DNN takes this information to perform the training. During the prediction phase, only the wearable sensor data without a label is sent, and the DNN responds with the most probable activity. By performing a set of 100 experiments, we obtained a mean average time of 10 ms: in this sense, this is a good result considering that the system is expected to be used in real-time.

After the training process using the dataset we presented in "[Data management](#)" section, the model has achieved an accuracy of 99.56% on the test set. Once we trained the model, we used a Keras functionality that allows to store it in a file allowing to load it without re-training the model every time. In particular, the trained model serialization has a dimension of 54 KB which is a reasonably tiny footprint if we consider the fact that it has to be exported to a Raspberry device, to be used then for the inference process.

In Fig. 7, we show the learning curves obtained by computing the train and the validation losses, as the training epochs increase. As we can observe, both curves converge to a near-zero value, thus demonstrating that the model has learned the relationship between the input and the output correctly. In Fig. 8, on the other hand, we also show how the performance learning curves that represent the training and validation accuracy grow during epochs and this helps diagnose learning dynamics.

Comparison with other machine learning approaches

In this subsection, we present more extensive results of our system, and we compare it with other popular machine learning approaches. To measure the performance of our



classification algorithm, we used the Confusion Matrix (CM) that we show in Fig. 9. CM is a popular metric that shows the obtained results under the form of a table, where predicted classes are represented in the columns of the matrix, whereas actual classes are in the rows. Such a matrix contains four values for each class: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). ‘True’ or ‘False’ indicates if the classifier predicted the class correctly, whereas ‘Positive’ or ‘Negative’ indicates if the classifier predicted the desired class. The first two values indicate the number of correctly classified samples while the last two refer to the number of wrong samples by the algorithm.

In particular, starting from these values, it is possible to compute the *Precision*, *Recall* (or *Sensitivity*), and *F1-score*, which three indicators are used to better evaluate the performance of a classification algorithm, since they convey more information than

Table 4 Classification metrics report obtained from the Confusion Matrix applied to our model

DNN performance metrics				
Label	Precision	Recall	F1-Score	Support
0 (Supine)	0.99	1.00	1.00	330
1 (Prone)	1.00	1.00	1.00	204
2 (Right)	1.00	1.00	1.00	202
3 (Left)	1.00	1.00	1.00	139
4 (Sitting)	0.99	1.00	1.00	341
5 (Movement)	1.00	0.93	0.97	91
Avg/total	1.00	1.00	1.00	1307

accuracy alone. Specifically, Precision indicates the ratio of labels correctly predicted (TP) on all those that the model classified as positive (TP and FP). More intuitively, the Precision metric indicates the accuracy of the algorithm in classifying the correct labels:

$$Precision = \frac{TP}{TP + FP}. \quad (5)$$

Recall, also defined as *sensitivity*, indicates the share of accurate estimates on the total of the positives. It is computed as the ratio between the TP and the total number of positives samples. Usually, recall is used when the goal is to minimize the number of false negatives (FN), and obtained by using the following formula:

$$Recall = \frac{TP}{TP + FN}. \quad (6)$$

Many times the variability of these two indices is very high, in fact depending on the number of TP, FP, and FN these two metrics can be very different, making a comparison between two algorithms hard. In such a context, it is usually preferred to use the F1-score index which takes into account both the precision and the recall, and provides an immediate estimate on the accuracy of the model, with a value between 0 and 1, where 1 means that the algorithm was good at correctly classifying the samples on the test set, while 0 means that the model was wrong to classify all the samples.

F1-score can be represented as the harmonic mean of precision and recall:

$$F1\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (7)$$

In Table 4 we show the metrics computed from the CM of our proposed model.

Confusion Matrix converts the labels (i.e., Supine, Prone, Right, Left, Sitting, Movement) into numerical values to perform the measurements. Considered that our multi-class classification problem consists of six classes, each class is numbered with a value which ranges between from 0 to 5. The header Avg/total represents the 'weighted' average performed on each column of the confusion matrix (i.e., Precision, Recall, and F1-score) where the "Support" column represents the weights.

In particular, our model obtained very good results in terms of precision, recall, and F1-score for all the six classes, which proves its quality. To compare the performance

Table 5 Performance metrics comparison with other classification models

Performance metrics				
Classification model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
DNN	0.9956	1.00	1.00	1.00
SVM	0.9932	0.99	0.98	0.99
RF	0.8280	0.70	0.82	0.75

of our proposed approach with other models, we trained and tested other algorithms using the same dataset and in particular SVMs and RF algorithms. The methods that are based on SVMs have been usually suggested as an alternative to simple NN and DNN, and they are very used in the classification tasks with excellent generalization results. In such a context, we used a model based on SVMs with a *Radial Basis* kernel and a penalty term equal to 1.0. SVMs were able to generalize very well the data reaching an accuracy of 99.32% on the test set. Even if the result is very good and comparable to ours, it is worth to mention that SVMs need to perform a 1-vs-1 classification which requires to train $N \cdot (N - 1)/2$ (where N is the number of classes) intermediate models in order to perform the actual classification. In this sense, during the inference process, the SVMs pass the data to each intermediate classifier returning as a final result the class which obtained the higher score accuracy. On the contrary, our DNN approach does not need to train intermediate models, since it can perform a multi-class classification with a single model, thus resulting faster, in terms of inference time, while saving space. In addition, we also used the RF classifier by setting 100 estimators and applied to the same dataset achieving an accuracy of 82.80%.

As we can see in the summary Table 5, our proposed approach provided significant performance across all metrics for our dataset when compared with other popular two machine learning models.

Model deployment

Last part of the system is a web server that accumulates patients data and shows it using the web interface. We developed a web interface that medical staff will use to show live data on the screen and see the patient risk of developing PUs. To do that, we developed different PHP files that make a simple reading from the database internal to the web server containing for each patient the position history predicted by the DNN. If the DNN predicts the same position for an extended amount of time, the system will send an alarm signal meaning that the patient has to change the position as soon as possible to avoid PU formation.

The most important pages are shown in Figs. 10 and 11 which indicate the list of patients in a specific room with the number of the bed, the last position/activity and the level of risk of developing a PU associated with the patient. When the user clicks on the “show activity” button, a new page appears showing the patient historical activities. The system can generate an alert that warns the medical staff that a specific patient has remained too long in the same position. In this sense, the level of risk of PUs formation is estimated based on the amount of time the patient remained in the same position, together with a medical evaluation. The level of risk is highlighted with three colours: the

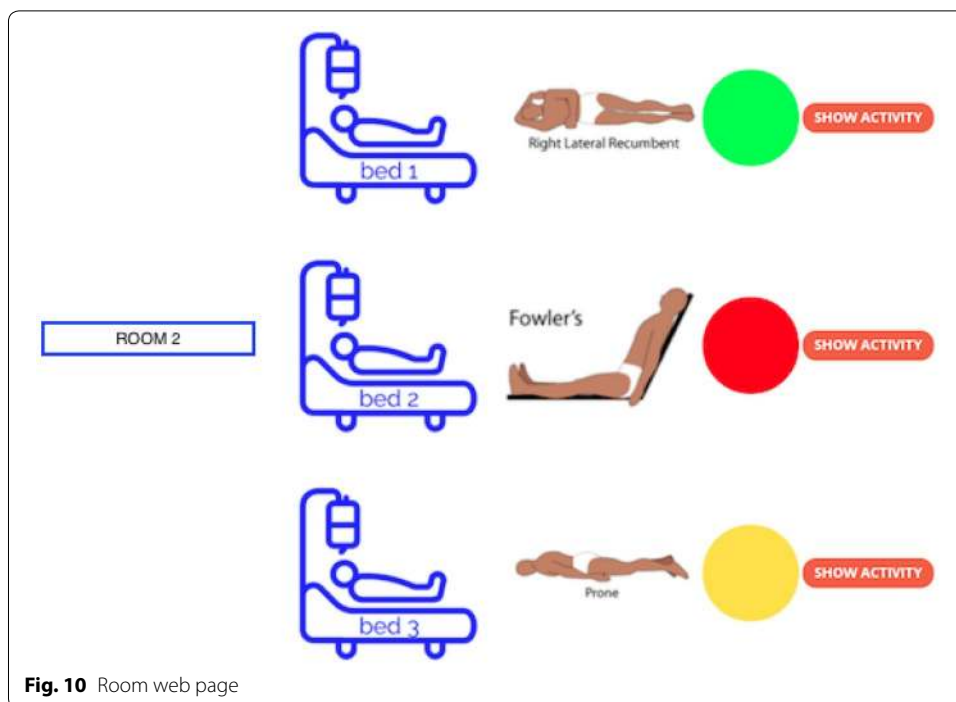


Fig. 10 Room web page

red colour means that the patient has been for more than two hours in the same position which means a high level of risk for the PU formations. The yellow colour means that the patient has been for half an hour, which indicates a mid-level of risk for PU formation. The green colour means that he/she has recently been in that position. The circle, associated with the patient, will change colour according to his/her stillness, as specified in the guidelines of the risk assessment for PUs. These guidelines have been provided to us by the medical staff. In particular, the guides say that, after two hours of stillness, PUs could develop and therefore the patients should change their position in order to prevent it. The obtained results demonstrate the feasibility of the technique and suggest that the use of wearable devices can be considered a viable solution in order to avoid PU formation.

Conclusions

In this paper, we presented a deep learning approach to determine the motion activities of patients and address the problem related to the generation of PUs when the skin is under constant pressure for a long time. In such a context, we demonstrated that wearable technology could be considered a valid solution to address this kind of problem by implementing a non-invasive low-cost human-centric system which demands minimal hardware costs and setup. We designed a deep learning algorithm and tested it on a real scenario by evaluating system behaviour on several patients to verify the correctness and feasibility of the proposed approach. Results are encouraging and demonstrate that deep learning techniques give better performance compared to other machine learning techniques like SVMs and RF, and can be considered reliable support for patient health also in the domain of PUs prevention. Future works will be devoted to further extending the analysis with a comparison featuring other solutions, both in terms of research and

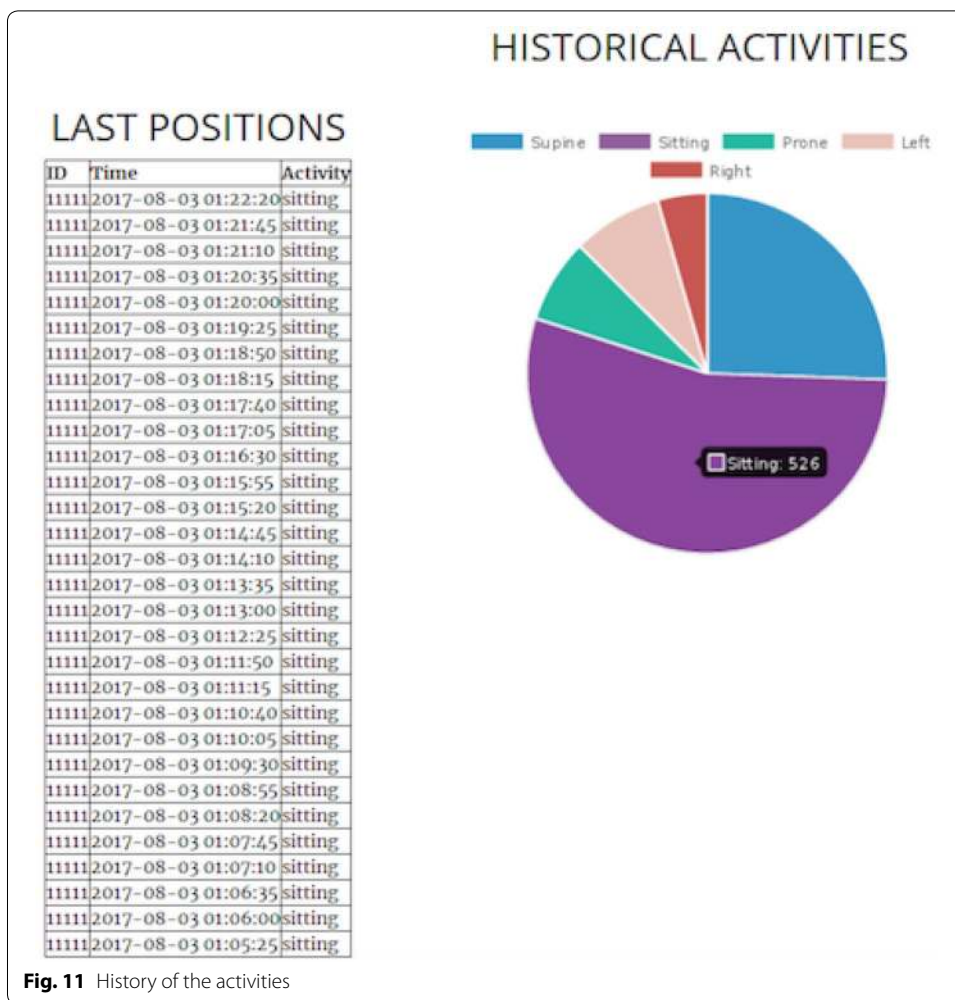


Fig. 11 History of the activities

industrial products, and to the investigation of novel techniques aimed at improving the overall system performance. About the latter, the analysis will focus on overall system responsiveness, real-time behaviour for data exchange, and performance of core system components, in particular, battery life of wearable sensors.

Acknowledgements

Not applicable.

Authors' contributions

GC devised the concept behind the work. GC, DB and FDV have taken part in data curation. GC, GM and FDV have taken care of formal analysis. GC, DB, GM, and FDV have took part in the investigation. GC, FDV and GM have taken care of the methodology. GM, GC and FDV have taken part of resources and software. AP, DB and GM have contributed to the supervision of the whole process. FDV, GM and GC have contributed to the validation. GC, GM and AP have taken care of the Visualization work. GC, GM, FDV and DB took part in Writing the original draft; GM and DB took part in review and editing duties. All authors read and approved the final manuscript.

Funding

The authors declare that they received no funding for this work.

Availability of data and materials

The dataset supporting the conclusions of this article is available in the Zenodo repository (<https://doi.org/10.5281/zenodo.3351545>).

Competing interests

The authors declare that they have no competing interests.

Author details

¹ Department of Engineering, University of Messina, Messina, Italy. ² CINI, National Interuniversity Consortium for Informatics, Rome, Italy.

Received: 28 July 2019 Accepted: 18 January 2020

Published online: 03 February 2020

References

- Abadi M et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>. Accessed 3 Dec 2019
- Amft O (2018) How wearable computing is shaping digital health. *IEEE Pervasive Comput* 17(1):92–98. <https://doi.org/10.1109/MPRV.2018.011591067>
- Barsocchi P (2013) Position recognition to support bedsores prevention. *IEEE J Biomed Health Inform* 17(1):53–59. <https://doi.org/10.1109/JTTH.2012.2220374>
- Bruneo D, Distefano S, Longo F, Merlino G, Puliafito A (2018) I/Ocloud: adding an IoT dimension to Cloud infrastructures. *Computer* 51(1):57–65
- Cao C, Liu F, Tan H, Song D, Shu W, Li W, Zhou Y, Bo X, Xie Z (2018) Deep learning and its applications in biomedicine. *Genom Proteom Bioinform* 16(1):17–32. <https://doi.org/10.1016/j.gpb.2017.07.003>
- Chang M, Yu T, Luo J, Duan K, Tu P, Zhao Y, Nagraj N, Rajiv V, Priebe M, Wood EA, Stachura M (2018) Multimodal sensor system for pressure ulcer wound assessment and care. *IEEE Trans Ind Inform* 14(3):1186–1196. <https://doi.org/10.1109/TII.2017.2782213>
- Choi S (2016) Understanding people with human activities and social interactions for human-centered computing. *Hum-centric Comput Inf Sci* 6(1):66:1–66:10. <https://doi.org/10.1186/s13673-016-0066-1>
- Chollet F et al (2015) Keras. <https://keras.io>. Accessed 5 Dec 2019
- Cui Y, Shi G, Liu X, Zhao W, Li Y (2015) Research on data communication between intelligent terminals of medical internet of things. In: 2015 international conference on computer science and applications (CSA), pp 357–359. <https://doi.org/10.1109/CSA.2015.39>
- Dhillon MS, McCombie SA, McCombie DB (2012) Towards the prevention of pressure ulcers with a wearable patient posture monitor based on adaptive accelerometer alignment. In: 2012 annual international conference of the IEEE engineering in medicine and biology society, pp 4513–4516. <https://doi.org/10.1109/EMBC.2012.6346970>
- Díaz C, García-Zapirain B, Castillo C, Sierra-Sosa D, Elmaghraby A, Kim PJ (2017) Simulation and development of a system for the analysis of pressure ulcers. In: 2017 IEEE international symposium on signal processing and information technology (ISSPIT), pp 453–458. <https://doi.org/10.1109/ISSPIT.2017.8388686>
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press. <http://www.deeplearningbook.org>. Accessed 6 Dec 2019
- Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. *Future Gener Comput Syst* 29(7):1645–1660
- Hayn D, Falgenhauer M, Morak J, Wipfler K, Willner V, Liebhart W, Schreier G (2015) An ehealth system for pressure ulcer risk assessment based on accelerometer and pressure data. *J Sens* 2015:106,537:1–106,537:8
- Hu F, Xie D, Shen S (2013) On the application of the internet of things in the field of medical and health care. In: 2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing, pp 2053–2058. <https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.384>
- Huang J, Lin S, Wang N, Dai G, Xie Y, Zhou J (2020) TSE-CNN: a two-stage end-to-end cnn for human activity recognition. *IEEE J Biomed Health Inform* 24(1):292–299. <https://doi.org/10.1109/JBHI.2019.2909688>
- Kaşıkcı M, Aksoy M, Ay E (2018) Investigation of the prevalence of pressure ulcers and patient-related risk factors in hospitals in the province of Erzurum: a cross-sectional study. *J Tissue Viability* 27(3):135–140. <https://doi.org/10.1016/j.jtv.2018.05.001>
- Longo F, Bruneo D, Distefano S, Merlino G, Puliafito A (2016) Stack4Things: a sensing-and-actuation-as-a-service framework for IoT and Cloud integration. *Ann Telecommun.* <https://doi.org/10.1007/s12243-016-0528-5>
- Mliki H, Bouhlel F, Hammami M (2020) Human activity recognition from UAV-captured video sequences. *Pattern Recogn* 100:107140. <https://doi.org/10.1016/j.patcog.2019.107140>
- Nisar H, Malik AR, Asawal M, Cheema HM (2016) An electrical stimulation based therapeutic wearable for pressure ulcer prevention. In: 2016 IEEE EMBS conference on biomedical engineering and sciences (IECBES), pp 411–414. <https://doi.org/10.1109/IECBES.2016.7843483>
- Nweke HF, Teh YW, Mujtaba G, Alo UR, Al-garadi MA (2019) Multi-sensor fusion based on multiple classifier systems for human activity identification. *Hum-centric Comput Inf Sci* 9(1):34. <https://doi.org/10.1186/s13673-019-0194-5>
- Sen D, McNeill J, Mendelson Y, Dunn R, Hickie K (2018) A new vision for preventing pressure ulcers: wearable wireless devices could help solve a common-and serious-problem. *IEEE Pulse* 9(6):28–31. <https://doi.org/10.1109/MPUL.2018.2869339>
- Shrestha A, Li H, Fioranelli F, Le Kernec J (2019) Activity recognition with cooperative radar systems at C and K band. *J Eng* 2019(20):7100–7104. <https://doi.org/10.1049/joe.2019.0559>
- Takano M, Ueno A (2019) Noncontact in-bed measurements of physiological and behavioral signals using an integrated fabric-sheet sensing scheme. *IEEE J Biomed Health Inform* 23(2):618–630. <https://doi.org/10.1109/JBHI.2018.2825020>
- Tsai C, Li C, Lam RW, Li C, Ho S (2020) Diabetes care in motion: blood glucose estimation using wearable devices. *IEEE Consumer Electron Mag* 9(1):30–34. <https://doi.org/10.1109/MCE.2019.2941461>

26. Wang TY, Chen SL, Huang HC, Kuo SH, Shiu YJ (2011) The development of an intelligent monitoring and caution system for pressure ulcer prevention. In: 2011 international conference on machine learning and cybernetics, vol 2, pp 566–571. <https://doi.org/10.1109/ICMLC.2011.6016779>
27. Wåhslén J, Lindh T (2017) Real-time performance management of assisted living services for bluetooth low energy sensor communication. In: 2017 IFIP/IEEE symposium on integrated network and service management (IM), pp 1143–1148. <https://doi.org/10.23919/INM.2017.7987452>
28. Yousefi R, Ostadabbas S, Faezipour M, Nourani M, Ng V, Tamil L, Bowling A, Behan D, Pompeo M (2011) A smart bed platform for monitoring amp; ulcer prevention. In: 2011 4th international conference on biomedical engineering and informatics (BMEI), vol 3, pp 1362–1366. <https://doi.org/10.1109/BMEI.2011.6098589>
29. Zhong Z, Li Y (2016) A recommender system for healthcare based on human-centric modeling. In: 2016 IEEE 13th international conference on e-business engineering (ICEBE), pp 282–286. <https://doi.org/10.1109/ICEBE.2016.055>
30. Zhu J, San-Segundo R, Pardo JM (2017) Feature extraction for robust physical activity recognition. *Hum-centric Comput Inf Sci* 7(1):16. <https://doi.org/10.1186/s13673-017-0097-2>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
