Agent-based simulation of smart beds with Internet-of-Things for exploring big data analytics

Iván García-Magariño, Raquel Lacuesta, and Jaime Lloret, *Senior Member, IEEE*

***Abstract*—Internet-of-things can allow healthcare professionals to remotely monitor patients by analyzing the sensors outputs with big data analytics. Sleeping conditions are one of the most influential factors on health. However, the literature lacks of the appropriate simulation tools to widely support the research on the recognition of sleeping postures. The current work proposes an agent-based simulation framework to simulate sleeper movements on a simulated smart bed with load sensors. This framework allows one to define sleeping posture recognition algorithms and compare their outcomes with the poses adopted by the sleeper. This novel presented ABS-BedIoT simulator allows users to graphically explore the results with starplots, evolution charts, and final visual representations of the states of the bed sensors. This simulator can also generate logs text files with big data for applying offline big data techniques on them. The current approach is illustrated with an algorithm that properly recognized the simulated sleeping postures with an average accuracy of 98%. This accuracy is higher than the one reported by an existing alternative work in this area.**

***Index Terms*—agent-based-simulation, big data, Internet-of-things, multi-agent systems, smart bed**

I. Introduction

Internet-of-Things (IoT) has allowed people to collect and analyze information from many environments, devices and objects integrated in common daily activities [1]. The objects with IoT normally have sensors that produce huge amounts of data that can be heterogeneous and imprecise. The big data analytics field provides the necessary tools and mechanisms for analyzing these data [2].

IoT has been used for both outdoor objects such as the groundwater sensors [3] and indoors objects like house hold appliances [4] and indoor self-location devices [5] . IoT has also been used for monitoring patients in the healthcare context [6]. In particular, IoT makes it possible the remote monitoring of patients. In healthcare, the wearable IoT sensors has become especially popular with diverse applications like the recent one about the identification and control of the spread of the Chikungunya virus [7]. A recent survey suggests that IoT can bring great advances in nursing care in the future [8].

Most people dedicate from 5 to 9 hours for sleeping every day. Some studies show that the duration and quality of sleep are related to health outcomes [9]. For example, short and long sleepers have greater risks of getting certain diseases than normal length sleepers. Sleeping postures are also related with some factor risks in some diseases such as sleep apnea and pressure ulcers. In particular, [10] presented a dense pressure sensitive bedsheet to monitor patients, and performed an analysis of the sleeping postures from the collected data. Their method presented a reliable posture recognition with an accuracy of 83%. In addition, there is also a US patent that proposed a system for sensing sleeping [11]. Nonetheless, the literature lacks of a software simulation framework that allows researchers to test new sleeping posture recognition algorithms without the need of using the aforementioned expensive equipment, to the best of the authors' knowledge.

Multi-agent systems (MASs) have proven to be especially useful for implementing healthcare systems in which several entities are coordinated. For example, [12] provided a MAS for supporting a collaborative wireless sensor network for health monitoring in a large structure. In addition, MASs can normally implement communications through Internet, as one can observe in the methodology for integrating MASs and web services [13]. More concretely, agent-based simulators (ABSs) are a specific kind of MASs that has been useful for simulating different health indicators. For instance, ABS-MindHeart [14] is an ABS that simulates the evolution of the heart-rate variability of a group of mindfulness meditators. In addition, ABSEM [15] is an ABS that simulates the emotions and the bodily sensation maps of some meditators following specific mindfulness interventions.

In this context, this work presents a framework for simulating sleeping postures for promoting and facilitating the research area about sleeping posture recognition through smart beds with sensors and IoT. In particular, we developed an ABS that simulates different kinds of sleepers in a smart bed with load sensors. This simulator is called ABS-BedIoT, and its underlying framework provides support for the development of sleep posture recognition algorithms. The ABS simulates these algorithms graphically showing the evolution and the final resulting outcomes. The ABS also generates logs with big data about the simulated signals of the smart bed sensors with different kinds of sleepers, so that researchers can explore big data analytics for sleep posture recognition. The current approach is illustrated with a sleeping posture recognition algorithm.

The current approach can be useful for advancing the measurement of human activities in ambient assisted living (AAL) through IoT [16]. In fact, the proposed approach could be used for integrating smart beds in smart communication architectures like the one proposed in [17]. The smart bed data

I. García-Magariño and R. Lacuesta are with the Department of Computer Science and Engineering of Systems of the Universtiy of Zaragoza, Teruel, 44003, and with the *Instituto de Investigación Sanitaria Aragón*, Zaragoza, Spapin. e-mails: ivangmg@unizar.es, lacuesta@unizar.es

J. Lloret is with the Instituto de Investigacion para la Gestión Integrada de Zonas Costeras, Universitat Politecnica de Valencia, Valencia, Spain. e-mail: jlloret@dcom.upv.es

Manuscript received April 19, 2005; revised August 26, 2015.

could be collected through a wireless sensor network with the algorithm introduced by [18] or with a suppression-based data collection approach [19]. The security could be guaranteed by a certificate revocation algorithm for removing malicious nodes from the IoT system [20].

The remainder of the paper is organized as follows. The next section introduces the related works highlighting some gap of the literature. Section III presents the novel ABS that covers this gap by simulating sleepers in smart beds. Section IV assesses the current approach by comparing the estimated values and the observed ones, in both the final results and the ones in each evolution step. Section V discusses the most relevant findings of the current work. Finally, section VI mentions the conclusions and depicts some future lines of research.

II. Related work

IoT has been considered relevant in health systems. For instance, [21] surveyed the state of the art about the smart objects in the context of IoT applied to health care systems providing cyber-physical smart (CPS) pervasive frameworks. They highlighted the relevance of security and big data analytics in this context. In addition, [22] proposed an open platform that combined intelligent medicine boxes wearable bio-medical sensor devices connected with Internet. Their approach with IoT was proposed for improving the user experience and the proper real-time monitoring of patients.

Several works of the literature have proposed to use smart beds for different purposes. These works introduced different kinds of sensors and information related to smart beds [23] proposed a smart bed for monitoring the physiological factors of patients in a non-intrusive way. This smart bed monitored the respiration, the heart rate, and the movement of patients with optic sensors. In addition, [24] proposed a contact-free way of monitoring respiration for smart beds, by capturing their movements. [25] suggested to use smart beds for decreasing stress and increase well-being in healthy people. This bed would allow business men to be less stress in the morning, by remembering their preferred sound, light and temperature settings to wake up them.

In the Consumer Electronic Show 2014 (CES'14), some smart home devices were showcased including the SleepNumber smart bed. This bed was mentioned as a possibility to be included in the networking-based smart home of [26]. They proposed a networking approach for making smart devices of homes communicate data autonomously, in order to facilitate the assisted-living and the comfort of users. In addition, [27] introduced a software defined smart home, which was based on the principles of software-defined networks (SDNs) for communicating smart home devices. They also mentioned the SleepNumber smart bed when introducing the relevance of the possible applications of their work.

The sleeping postures has been recognized previously in the work of [10] with a sensitive bedsheet. This included a high amount of pressure sensors in order recognize the posture of the sleeper. They presented an algorithm that obtained 83% of accuracy in detecting the postures of sleepers. They detected the sleeper postures from a set of six postures: right/left fetus postures, right/left log postures, supine position and prone position.

Therefore, several works have mentioned smart beds as a possible piece of furniture that can include sensors and can be interconnected with Internet or home networks. Most of these were different from the current work as either they did not recognize sleeper postures or they used a different sensors mechanism. The most similar work about the recognition of sleep postures is probably the one proposed by [10], which estimated the postures of the sleeper from the pressure sensors of a bedsheet. However, none of this works provided a simulation framework for assessing sleeping posture recognition algorithms without needing costly equipment as their proposed sensitive bedsheet.

Finally, it is worth highlighting that the bed rest has been considered relevant for the well-being of people, and this is conditioned by the sleep postures. As an example, the PhD of [28] analyzed the effects of mattresses and pillow designs on the sleep quality, spinal alignment and pain reduction. In some cases, the bed rest goes beyond the sleeping hours, like in the cases of pregnant women or some kinds of patients, and in these cases they may need communication with Internet so their physical and psychological conditions are followed. [29] proposed to make pregnant women communicate through mail so they can share their state about their self-perceptions. The current work goes beyond this communication, and proposes to advance towards the possibility that the bed itself communicates the states of the corresponding person by sensing the way they lay on the bed.

III. ABS-BedIoT

The goal of ABS-BedIoT is to provide an IoT simulation testbed for experiencing big data analytics algorithms in the context of smart beds. ABS-BedIoT simulates a bed with a grid of load sensors as a mechanism for detecting certain sleeping poses and the amount of movement. In the real world, this detection could be helpful for detecting injuring poses in critical patients or chronic ones. Nurses may need to monitor the sleeping poses of some of these patients.

In a first layer, ABS-BedIoT provides the basics mechanism for simulating the poses of sleeper. In order to provide different testbed scenarios, we developed three different kinds of sleepers based on stochastic behaviors. It also provides a basic smart bed with sensors that detect the weight of the sleeper in certain positions of the bed.

In a second layer, ABS-BedIoT includes an analyzer of the information of the sensors in order to detect the sleeper's poses. This is an initial proposal and is intended to be extended by other researchers to experience different sleeping pose detection algorithms. In fact, ABS-BedIoT has the possibility of generating log files with all the information of sensors step-by-step during simulations. Thus, sleeping pose detection algorithms can be developed and tested as separate tools by just processing the text files with big data generated by ABS-BedIoT.

ABS-BedIoT has been developed as an ABS, in order to make it possible to model the behavior of the different components of the bed as autonomous and reactive entities. This paradigm also allowed as to simulate the sleeper as a separate agent that can adopt different behaviors.

Section III-A presents the design of the ABS introducing the different agent types. Section III-B introduces the underlying concepts for representing an sleeper and its postures. Section III-C indicates the representation of the smart bed and the simulation of its load cells. Section III-D presents an algorithm for recognizing sleeping postures from the signals of the smart bed sensors. Section III-E describes the implementation of the behaviors of several sleeper types. Section III-F introduces the user interface (UI) of ABS-BedIoT. Finally, section III-G presents the generation of logs with big data so that researchers can explore the application of big data analytics to these logs.

A. ABS design

The presented application has been developed following the Process for developing Efficient ABSs (PEABS) [30]. We selected that process as it allowed us to achieve high levels of performance, which is necessary when dealing with the generation of logs that quickly increase their size. We decided to use the Unity engine, since it allows one to build multi-platform mobile and desktop applications, and eases the development of visual interfaces.

In agent-based modeling, one of the most relevant phases is the definition of the agent types with their goals. In particular, we defined the following agent types aiming at increasing the cohesion of operations within each agent and reducing the coupling between different agents as recommended by the principles of agent-oriented architectures [31]:

- *Sleeper agent*: This agent simulates a person that is sleeping in the smart bed.
- *Weight Sensor agent*: This agent type simulates a load cell that weights the pressure in a given point of the bed.
- *Bed agent*: This agent simulates a smart bed, and manages the weight sensor agents.
- *Observer agent*: It observes the sleeper and records its states.
- *Analyzer agent*: It analyzes the information of the sensor agents using big data analytics.
- *Stochastic sleeper agent*: This agent extends the sleeper agent, and uses nondeterministic behaviors to change its pose.
- *Bad sleeper agent*: It simulates the behavior of a bad sleeper by extending the stochastic sleeper agent.
- *Restless Sleeper agent*: It impersonates the behavior of a restless sleeper person by extending the stochastic sleeper agent.
- *Healthy sleeper agent*: It represents a sleeper with a very long time of deep sleeping, and barely changes its pose.

Figure 1 shows the functional block diagram of ABS-BedIoT, where one can observe the main steps of the system. This diagram summarizes the simulation process for each simulation step. It briefly introduces the main action of each agent type. The next subsections will further introduce these actions.

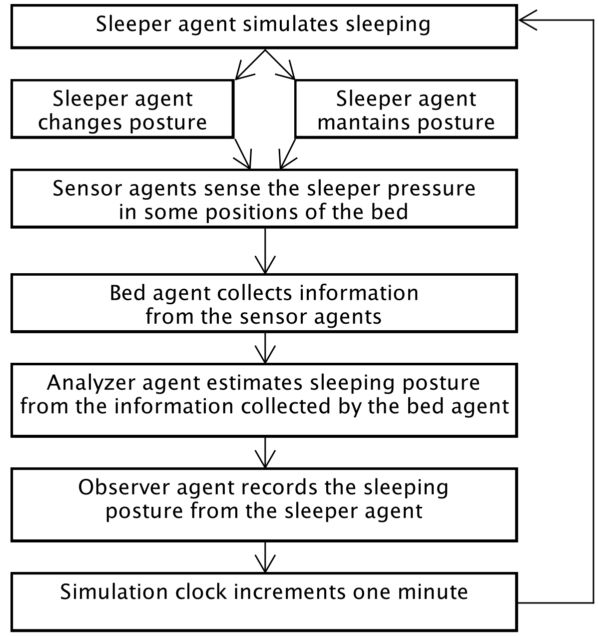

Fig. 1. Functional block diagram of ABS-BedIoT

B. Sleeper basics

The sleeper agent has a position in the bed associated with its body center. In this approach, the center of the body is considered to be the hips. This center is initialized as the center of the bed, since the sleeper is assumed to start the night in the center of the bed. The center of the bed is located in the (0,0) point.

In the current approach, the sensors ignore the weight of the limbs, so the representation of the sleeper is focused on the body and the head. In particular, the representation of the sleeper considers the following body parts as points of interest: the hips, the spine, the shoulders and the head. The center of each body part is associated with a position. In the case of shoulders, the position is the middle place between both shoulders.

In order to determine the position of each boy part, we used a set of local transformations as commonly done in most space representations in general [32], and more concretely also in the representation of some MASs [33]. In this particular case, we consider two-dimensional (2D) points since the sleeper lays in the bed, which is only a plane. First, a list of right positions determines the position locally arranged from the body center when the user is straight, which is considered as the normal/right pose. Second there is a list of deviations that determine the offsets from the right positions as 2D vectors. Therefore, the following spatial sum calculates the actual position of a body part from the corresponding 2D vectors:

$$\overrightarrow{P} = \overrightarrow{C} + \overrightarrow{R} + \overrightarrow{D} \quad (1)$$

where $\overrightarrow{P}$ is the position of the body part, $\overrightarrow{C}$ is the center of the sleeper, $\overrightarrow{R}$ is the normal displacement of the body part from the center, and $\overrightarrow{D}$ is the deviation from the right position.

Regarding the orientation of the sleepers' body, this approach considers two possibilities as many other works (see an example in [10]). First, the sleeper can be laid on one side. This works denotes this pose as a "lateral" pose. Otherwise, the user can be in a flat pose, meaning that it lays on its back or its stomach. In this case, its body will occupy a wider area of the bed. This approach refers these poses as 'frontal" poses. The sleeper contains a boolean flag that indicates whether the sleeper is in a lateral position. Only in case that the sleeper is in a frontal position, it adds two points of interest in their representation by summing a horizontal offset symmetrically to both the right and the left of each body part. This is represented as lateral offsets and the new points of interest are calculated with the following formulas:

$$\overrightarrow{I_l} = \overrightarrow{P} + \overrightarrow{(-l,0)} \tag{2}$$

$$\overrightarrow{I_r} = \overrightarrow{P} + \overrightarrow{(l,0)} \tag{3}$$

where I_l and I_r are respectively the left and right points of interest for each body part, and l is the lateral offset for the given body part. Notice that the lateral offset is different for each body part, since for example shoulder normally occupy a wider area than hips.

The sleeper provides a method for providing the list of the points of interest given their pose. It is assumed that around certain distance of each body part, the sleeper has enough weight to be detected by a bed sensor.

Several poses were defined for the sleeper. Each of these poses is represented by a list of deviations for each body part. In this manner, it can adopt any pose, by replacing its current deviations with the list of deviations of the corresponding pose. Inspired by the literature, we defined several poses. The "neck bent" pose refers to a sleeper with a straight position of the body and the neck bent. The "body bent" represents a sleeper that has their body excessively bent. "Spine S" pose refers a pose in which the central points of interest have deviations in opposite directions conforming the shape of a "S" letter. "Normal pose" refers to an appropriate pose without the risk of getting injured, in which all the body parts are straightly aligned. The neck bent pose is inspired by works such as [34], and the the different body and spine poses are inspired by works like [35]. The body bent can be similar to the fetus posture [10] but with an excessive bend of the spine that can cause some injuries.

The sleeper agent has the appropriate methods to adopt different poses and body orientations, so that different subtypes of sleeper agent can be easily defined. On the whole, the posture of a sleeper is determined by two components, which are the body orientation (i.e. either frontal or lateral) and its actual pose (i.e. neck bent, body bent, spine S or normal).

C. Smart bed with sensors

The smart bed is composed of (1) the bed agent for representing the whole bed, and (2) several weight sensor agents that represent the load cells of the smart bed.

A load cell provides a signal that determines the weight/force that the sensor is sensing. The sensor agent should determine whether that weight is perceived as a presence of the sleeper. In order to simulate this, each sensor agent periodically requests the lists of positions of the body parts of the sleeper. It calculates the distance to each point of interest. If the distance to the closest point of interest is below certain s_t threshold, then the sensor agent is assumed to presence of the sleeper and outputs a positive signal represented with the one number. Otherwise, its signal indicates the absence of the sleeper with a zero. The activated mode of a sensor can be formally represented with the following condition:

$$s_a \Longleftrightarrow \exists \overrightarrow{X} \in P : |\overrightarrow{S} - \overrightarrow{X}| <= s_t \tag{4}$$

where s_a determines whether the sensor provides an activated signal of one, P is the set of points of interest including the body parts and the ones added for the frontal posture if any, $\overrightarrow{S}$ is the position of the sensor, and s_t is the sensor threshold as the minimum distance to a point of interest to detect the presence of the sleeper.

All the other agents and components that need to access the sensors' states, they do it through the bed agent. In the real world, the smart bed could transfer this information through Internet aligning with the approach of IoT. In more advanced versions, the bed can include the necessary software to pre-analyze these data and provide more reduced and meaningful information, such as the pose detected from the sleeper.

D. Detection of poses

In the current approach, the detection of poses can be either performed online or offline. The online method is carried by the analyzer agent. This agent detects the pose of the sleeper in each simulation iteration that represents each sleeping minute. For supporting the offline detection of poses, the presented simulator provides the possibility for generating logs of sensors states into text files, as it will be further discussed later in section III-G.

ABS-BedIoT has a default online detection mechanism. However, the underlying framework has been designed and programmed so that other researchers can easily incorporate new online mechanisms for analyzing sleeper poses. This can be achieved by extending the analyzer agent.

The analyzer agent incorporates methods to perform some low-level operations from which the detection algorithms can be built. Firstly, a method of the analyzer agent estimates the index of the row of sensors that is the nearest to a given body part. This converts the common Y position of a body part in the index of the row of sensors by calculating the Y position of each row of sensors. This can be robust to small variations of the location of the sleeper since there is a margin with the distance threshold. However, it is not robust for really strange positions of the sleeper agent. Notice, that sleepers normally move horizontal by rolling, but they rarely move up or down since they would get out of the bed soon, depending on their heights. This basic operation may be tuned in the future.

Another basic operation is to obtain the average index of the activated sensors of a given row. In particular, it represents the middle point of that row of sensors with the following formula:

$$m_y(b) = \overline{\{x \in [0, n) : s_{a,x,y}\}} \tag{5}$$

where m_y is the middle index of row y calculated as the average of the indexes of the activated sensors, b is the body part from which the y row is estimated, and $s_{a,x,y}$ determines if the sensor in the x column and y row is activated.

The analyzer agent defines the methods "Observe Lateral Posture" and "Observe Pose" to respectively determine the two components of the sleeper pose.

In the presented detection mechanism, the observation of the lateral posture is performed by selecting a body part and calculating the minimum and maximum indexes sensors of the corresponding row. Then, it calculates the distance between the positions of the two sensors and obtains an estimated width. A threshold width is set, and if the width is above this threshold, it is considered that the sleeper is in frontal position. Otherwise, it is considered that the sleeper is in lateral position. We tested the shoulders which are the ones with the highest variation of widths. However, the position of the head interfered in some lateral positions, and some posture orientations were wrongly classified. The hips were more accurate as long as the threshold was appropriate for the sleeper width. Notice that the variation of hips lengths is lower. The mechanism detected a lateral pose referred as the l_p condition output with the following formula:

$$l_p \iff w_p > l_{p,t} \tag{6}$$

where w_p is the width of sensed by the sensors of the row estimate for body part, and $l_{p,t}$ is the threshold for the sensing a lateral pose.

The second component of the sleeper pose was detected by considering the following conditions in which the similarity was determined by a given threshold. The following conditions are checked in order, and if one is true, then the estimations is returned without checking the subsequent conditions:

$$\begin{aligned} e_p =' normal' \iff & m_y('hips') \approx m_y('spine') \wedge \\ & m_y('spine') \approx m_y('shoulders') \wedge \\ & m_y('shoulders') \approx m_y('head') \end{aligned} \tag{7}$$

$$\begin{aligned} e_p =' neck\_bent' \iff & m_y('hips') \approx m_y('spine') \\ & \wedge m_y('spine') \approx m_y('shoulders') \wedge \\ & \approx m_y('shoulders') \not\approx m_y('head')) \end{aligned} \tag{8}$$

$$\begin{aligned} & e_p =' body\_bent' \iff \\ sign(m_y(&'shoulders') - m_y('spine')) = \\ & sign(m_y('head') - m_y('shoulders')) \end{aligned} \tag{9}$$

$$\begin{aligned} & e_p =' spine\_S' \iff \\ sign(m_y(&'shoulders') - m_y('spine')) \neq \\ & sign(m_y('head') - m_y('shoulders')) \end{aligned} \tag{10}$$

where e_p is the estimated pose, the sleeper poses and body parts are expressed between simple quotes, and the meaning of $m_y(b)$ is the previously defined for the estimated middle point of a body part.

Researchers can easily easily define new sleep posture detection algorithms by extending the class "Analyzer Agent" and overriding the "Observe Lateral Posture" and/or "Observe Pose" methods. These methods can access to the states of the sensors through the bed agent. Researchers can find implementation examples of these methods in the definition of the analyzer agent itself.

E. Sleeper's stochastic behaviors

We defined a generic stochastic sleeper's behavior based on nondeterministic decisions bases on probabilities as recommended by TABSAOND (a technique for developing ABS apps and online tools with nondeterministic decisions) [36]. The goal was to obtain a large variety of sleeper behaviors and that these are realistic, in the sense that the exact sleeping movements are different from night to night.

This generic stochastic sleeper agent implements this behavior, and has certain internal parameters. The assignment of different values to these parameters allows one to define a quite wide range of sleeper behaviors.

Each simulation iteration represents a sleeping minute, and this is simulated with the invocation of the "Live" method of the agent as recommended by PEABS. In each invocation, the sleeper agent sequentially (1) determines which kind of sleep is having, (2) indicates if it changes body orientation, (3) determines whether to change its pose, and finally (4) decides whether to change its location represented with its center position in the bed.

It is well known that people have different grades of sleep deepness [37]. They can have light sleep when they have just got to bed or before waking up in the morning. However, they normally have deep sleep in the middle hours of the night. In order to cover different kinds of sleepers, this agent has two parameters for determining the beginning time and the end time of the deep sleep measured with the number of minutes elapsed from the time the sleeper got to bed. The changes of all the next phases will rely on certain probabilities that are different between light and deep sleep.

In order to determine whether the sleeper changes its body orientation, we used the formula below following the proposal of TABSAOND for binary decisions:

$$d = \begin{cases} \text{change,} & \text{if } r \leq p_{o,sl} \\ \text{not change,} & \text{otherwise} \end{cases} \tag{11}$$

where r is a random number in the [0,1] interval, and $p_{o,sl}$ is the probability of changing the orientation for an sl intensity of sleep, which can be either light or deep sleep.

This work recommends that for any sleeper the probability of changing the orientation is lower or equal in the deep sleep than in the light sleep, in order to make coherent behaviors. The recommendation is formalized with the following formula:

$$p_{o,D} \leq p_{o,L} \tag{12}$$

where D is deep sleep and L is light sleep.

When the sleeper agent decides to change its position, it changes to the opposite orientation (i.e. from frontal to lateral or vice versa).

In each simulation iteration, the stochastic sleeper agent decides whether to change its pose with a similar stochastic binary decision, but this time based in a different probability denoted as $p_{ch,sl}$, where sl is the sleep intensity. If the agent decides to change its position, then it applies the formula proposed by TABSAOND for decisions between more than two options. This version adapted to the current case is the following:

$$d' = \begin{cases} ps_1, & \text{if } r \leq p_{ps1} \\ ps_2, & \text{if } p_{ps1} < r \leq (p_{ps1} + p_{ps2}) \\ ps_i, & \text{if } \sum_{j=0}^{i-1} p_{psj} < r \leq \sum_{j=0}^{i} p_{psj} \\ ps_N, & \text{otherwise} \end{cases} \tag{13}$$

where ps_1, ps_2, ... ps_N are respectively the neck bent, body bent, spine S and normal poses, and p_{ps1},p_{ps2}, ... $p_{ps(N-1)}$ are respectively the probabilities of adopting the poses ps_1, ps_2, ... $ps_{(N-1)}$, and the probability of ps_N is one minus the sum of the previous probabilities.

Notice that the previous formula is general enough to remain even when enlarging the set of poses considered in the proposed approach.

Finally, the center of the agent can be changed. This also adds variety to the possible combinations of activated sensors, and makes the simulations more realistic. A stochastic binary decision was defined with the probability p_{loc} to determine whether the sleeper changes its location in the bed in each simulation iteration. Then, another stochastic binary decisions was simulated to determine whether the sleeper moved to a normal position (i.e. around the center of the bed) with probability $p_{loc,norm}$ or whether it moved to a strange position of the bed (i.e considerably different from the center of the bed). In the latter case, the position $\overrightarrow{C}$ was selected with the following formula:

$$\overrightarrow{C} = \overrightarrow{(r_x, r_y)} \tag{14}$$

where r_x is a random number in the interval [x_{min},x_{max}], and r_y is a random number in the interval [y_{min},y_{max}]. The values x_{min}, x_{max}, y_{min} and y_{max} are internal limit parameters of the application that can be tuned.

The current approach proposes three different sleeper agents defined as extensions of the stochastic sleeper agent. In fact, the stochastic sleeper agent has some default values for all the probabilities and other internal parameters, and each extended sleeper agent just changes some specific probabilities in its constructor. The sleeper agent could also change any functionality. Three agents were defined for the following sleeper behaviors:

- *Bad sleeper*: This agent has a tendency in adopting the injuring poses of neck bent, body bent and spine S more frequently than normal. Therefore, the probabilities of these poses were increased up to $p_{ps1} = 0.30$, $p_{ps1} = 0.20$ and $p_{ps3} = 0.10$ respectively.
- *Restless sleeper*: This sleeper cannot rest in most of the night, thus the beginning and end of the deep sleeping time is changed respective to 360 and 420 min. In other words, it only rest in the 7th hour of the night, while the generic sleeper rests with deep sleep from the 2nd to the 7th hour. In addition, this agent has a higher probability of both changing the orientation and the pose when having a light sleep with respectively the probabilities of $p_{o,L} = 0.45$ and $p_{ch,L} = 0.35$.
- *Healthy sleeper*: The sleeper agent sleeps most of the night with deep sleep from 30 min after getting into the bed until 450 min of sleeping time. The probabilities of changing body orientation is very low for deep sleep with $p_{o,D} = p_{ch,D} = 0.03$ and also low for light sleep with probabilities $p_{o,L} = p_{ch,L} = 0.05$. The probability of adopting injuring poses was set to low values, using the probabilities $p_{ps1} = 0.05$, $p_{ps2} = 0.03$ and $p_{ps3} = 0.01$ for respectively neck bent, body bent, and spine S poses.

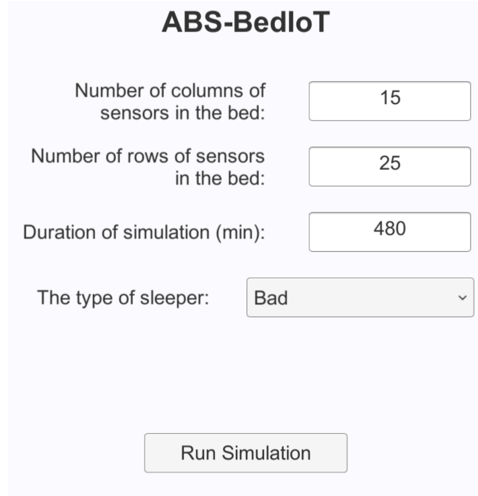

Fig. 2. UI for introducing the inputs of ABS-BedIoT

F. User interface

In the UI of ABS-BedIoT, the user can determine certain input parameters of simulations of beds, as one can observe in Figure 2. First, they can introduce the configuration of sensors indicating the numbers of columns and rows of the sensors. These are equally organized forming a grid of sensors. Then, the user can establish the duration of the simulation in minutes. Finally, they can select one of the existing types of sleepers from a dropdown list. The user can press the "Run simulation" button to start the simulation.

When the simulation has finished, the app presents a summary of the simulation results with a starplot. Figure 3 shows an example of this UI screen. In this screen, the user can observe the percentages of times in which the user was detected in each pose. The starplot has been selected for being an intuitive graphical representation, in which the user may understand the results at a glance. The scale is fixed to have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2764467, IEEE Access

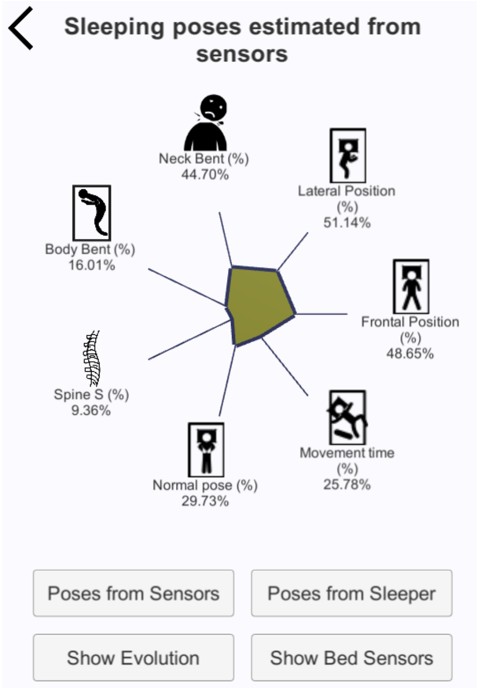

Fig. 3. Screen of the main results of the UI of ABS-BedIoT (taken from a simulation of the bad sleeper)

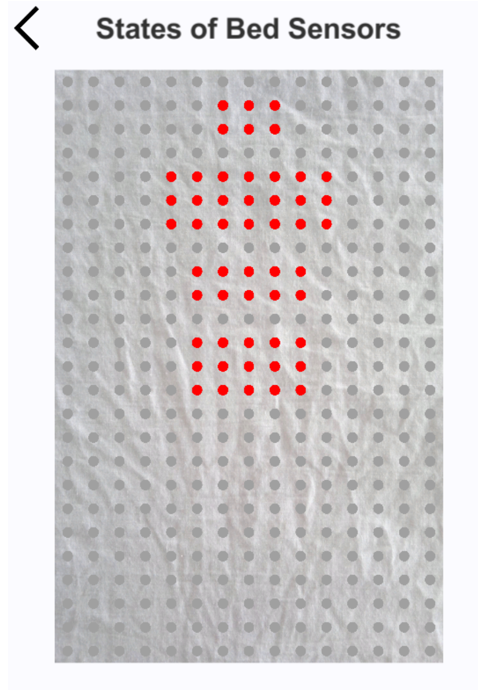

Fig. 4. Example of UI of the state of the bed sensors when the sleeper is in frontal pose

the maximum in 100%, even if not any measure achieves that value, so that the user can observe and compare different starplots in a meaningful way. It is worth reminding that front and lateral postures can be combined with any of the other poses, which are head blend, body blend, spine S and normal pose. Thus, the percentages can sum more than 100% since these poses can be combined.

Besides the time percentages of the sleeping poses, it also shows the time percentage that the sleeping is moving considering intervals of one minute.

This app can present both the (a) the time percentages of sleeping poses that the corresponding sleeper agent actually adopted, and (b) the the time percentages of sleeping poses estimated from the sensors. By default, the latter option is firstly presented to the user. However, the user can also request to see the real time percentages by pressing the “Poses from Sleeper” button. The user can switch between these two views by pressing that button and the “Poses from sensors” button. The screen title indicates which is the current view. In this way, the user can compare both results to assess the reliability of an algorithm for detecting emotions through sensors, and also the reliability of a grid configuration of sensors with certain numbers of rows and columns. In fact, the researchers can test different combinations of algorithms and grid configurations.

From the screen with the main results, the user can go to other two different screens of the UI containing respectively the simulation evolution and the final state of bed sensors. The user can access these screens respectively with the “Show Evolution” and ‘Show Bed Sensors” buttons.

In the screen of the simulation evolution, the user can observe the time percentage evolution of each kind of pose and the movement along the sleeping time measured in minutes, in a chart. This evolution is estimated from the sensors. However, the user can switch between this estimated evolution and the real one of the sleeper agent with two bottom buttons, and the change of view is reflected in the title of the screen. Section IV shows several examples of simulated evolutions.

When the user selects to see the bed sensors, they can observe a screen like the one shown in Figure 4. In this screen, the app shows the image of a bed with the sensors represented as circles. Considering certain threshold, each agent can either detect enough pressure from the sleeper or not do it. In the former case, the sensor is represented as a red circle. Otherwise, it is represented as a gray circle. In this particular example, the sleeper is in a frontal (i.e. non-lateral) posture, in a normal pose (meaning not any injuring pose). The sleeper is situated in the center of the bed, since their hips are around the center of the bed. Notice that this simulator only detects the pressure of the body, shoulders and head, and omits the detection of the limbs as these are less heavy.

G. Logs for exploring big data analytics

ABS-BedIoT is designed for letting researchers to explore big data analytics in the context of IoT, more specifically in the context of smart beds with Internet. This application allows one to simulate sleepers and record the states of sensors of the whole simulation considering one-minute intervals. This information is stored in a log file. The researcher can enable/disable this property internally in the “Experimentation” class.

The generated file contains the information of all the sensors for each minute tagged with the posture of the sleeper with their orientation (i.e. frontal or lateral) and their pose (neck bent, body bent, spine S or normal). Figure 5 shows an excerpt

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2764467, IEEE Access

ABSLogsSensorsRestlessSleeper.txt

```
Frontal Normal 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0
0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0
0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
|0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 1 1 1 1 1 0 0 0 0
0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 1 1 1 1 1 1 1 0 0
0 0 |0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 |0 0 0 0 1 1 1 1 1 1 1 0
0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0
0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 |
Lateral Normal 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0
0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
|0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0 0
0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0
0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0
0 0 0 0 |0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 |
Frontal NeckBent 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0
0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 1 1 1 1 1 0 0 0 0 0
|0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 |0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 |0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |0 0 0 0 1 1 1 1 1 1 1 0
0 0 0 |0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 |0 0 0 0 1 1 1 1 1 1 1
0 0 0 0 |0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 |0 0 0 0 1 1 1 1 0 0
0 0 0 0 0 |0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 |0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 |
```

Fig. 5. An excerpt of an example of the logs of sensors generated by ABS-BedIoT for the restless sleeper

of an example of this log file. The information of each one-minute interval is represented with a new line. The two-word posture tag is at the line beginning. Each sensor is represented with "1" if it is enough pressured by the sleeper or "0" otherwise. The sensors are represented row by row separated with the "|" character.

Most of the experiments have been performed with 15x25 grids of sensors and simulations of 480 min, as the one of the example file. The data amount generated by one simulation for each sleeper type was 1.11 MB in total. The data amount of 1,000 simulations for each sleeper type would be 1.08 GB in total. Notice that there are many nondeterministic decisions in each simulation, and consequently most of the simulations are different from each other. This amount of data can be useful for experiencing the techniques of big data analytics, and going further in the development of algorithms for analyzing big data from smart bed sensors.

IV. EXPERIMENTS

A. Step-by-step comparison of the simulation evolutions

In order to assess the evolutions of simulations, we compared the pose estimated from the bed sensors with the one directly observed from the sleeper for each iteration (i.e. each sleeping minute). Due to the nondeterministic behavior of the ABS, we executed 100 times each kind of sleeper. Each simulation had 480 iterations. Thus, 48,000 comparisons were performed. All the simulations were executed with a 15x25 grid of sensors.

TABLE I
ACCURACY (%) IN ESTIMATING POSES IN THE STEP-BY-STEP EVOLUTION

	Bad Sleeper	Restless Sleeper	Healthy Sleeper	*Total*
Frontal	97.37	97.93	98.07	97.79
Lateral	100.00	100.00	100.00	100.00
Neck Bent	99.84	96.51	98.30	98.22
Body Bent	96.07	99.79	99.12	98.33
Spin S	97.67	99.37	98.73	98.59
Normal	95.62	96.00	96.13	95.92
Average	97.76	98.27	98.39	98.14

We measured the accuracy of the estimator as the percentage of times its estimation matched the one directly observed from the sleeper agent. We analyzed separately the two factors of sleeping poses (first whether the sleeper was lateral positioned, and second which of the other poses the sleeper adopted). Table I presents the results, showing the accuracies for each pose and each sleeper type. This table also presents the averages for each sleeping pose and the averages for each sleeper type. One can observe that all the accuracies are within the 95-100% interval. The lateral sleeping posture is always detected with the highest accuracy, while the normal pose (referring to non-injuring poses) had the lowest accuracy even this is high (i.e. around 96% in average for the three sleeping types). Finally, the total accuracy average was 98.1%.

This result was considered a high accuracy average as it surpassed the sleep posture recognition average accuracy of a similar work of the literature by [10], who reported an average accuracy of 83.0%. However, the results of this comparative is not definite, as both works used different datasets and detected sleep postures from different sets.

In order to visually show that evolutions are generally quite similar, this work presents the estimated evolution and the one directly observed from the sleeper for the same simulation. In particular, Figure 6 shows the example of the evolution of the sleeping poses estimated from the sensors for a bad sleeper. Figure 7 shows the evolution of the sleeping poses directly observed from the sleeper for the same simulation. As one can observe, both evolutions are quite similar between each other although these are not exactly the same.

In order to further compare these two evolutions, we calculated the minimum, the maximum, the averages and the standard deviations (SD) of the values reported in the last 7 h of the simulated evolution. We discarded the first hour in this analysis and the next similar ones, since the time percentages were not representative as these were based on very few data. In fact, in the first minutes, these values were normally near 0% or 100% in most evolutions. The body orientation was very similar between estimated and observed values. The minimum and maximum of frontal orientation were respectively 42.3% and 59.1% in both evolutions. The average only varied from 53.0% to 52.9%, while the SD varied from 6.41% to 6.39%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2764467, IEEE Access

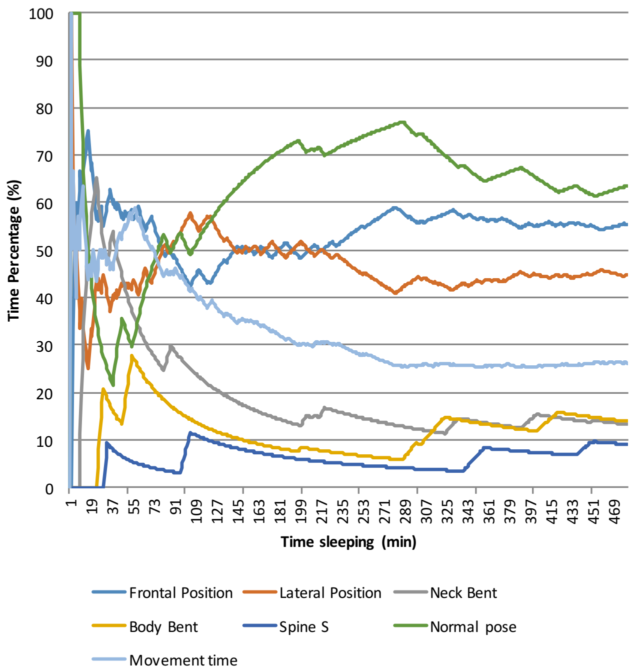

Fig. 6. Evolution of sleeping poses estimated from sensors for the bad sleeper

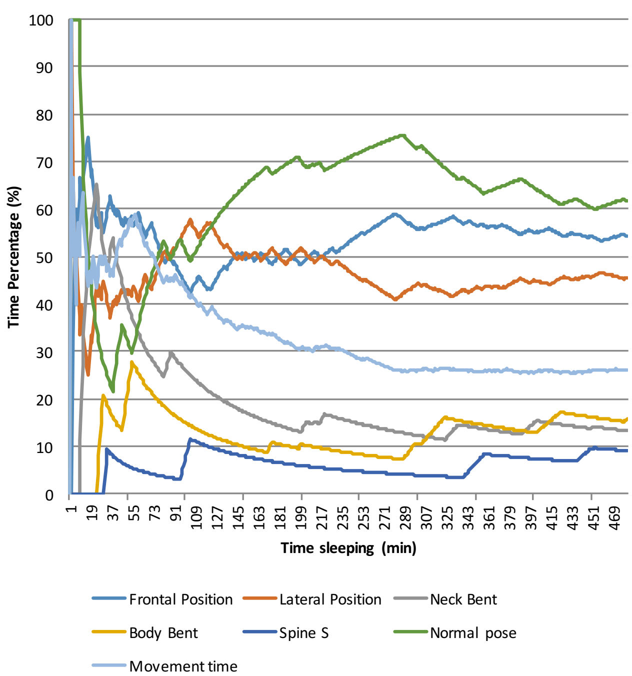

Fig. 7. Evolution of sleeping poses directly observed from sleeper in the same simulation of Figure 6

In most sleeping poses the averages, SDs, minimum and maximum were also very similar. The highest difference was obtained for the minimum of the body bent posture, in which the estimation was 5.92% and the observation was 7.32%.

TABLE II
TIME PERCENTAGES OF SLEEPING POSES AND MOVEMENT

	Observed (%)	Estimated (%)	Diff. Means (%)
Bad Sleeper			
Frontal	49.74 (4.25)	48.43 (4.69)	1.31
Lateral	50.06 (4.25)	51.34 (4.69)	-1.28
Neck Bent	30.03 (8.30)	29.11 (8.22)	0.92
Body Bent	19.42 (6.27)	21.94 (6.63)	-2.52
Spine S	8.56 (6.01)	8.79 (6.17)	-0.23
Normal	41.78 (2.18)	39.95 (8.50)	1.83
Movement	28.20 (2.18)	28.01 (2.13)	0.19
Restless Sleeper			
Frontal	50.51 (2.81)	49.46 (3.16)	1.05
Lateral	49.28 (2.81)	50.33 (3.16)	-1.05
Neck Bent	10.00 (2.61)	10.08 (2.64)	-0.08
Body Bent	9.87 (2.82)	12.07 (3.78)	-2.20
Spine S	4.92 (1.83)	5.65 (2.28)	-0.73
Normal	74.99 (3.81)	71.99 (4.52)	3.00
Movement	56.60 (2.18)	55.98 (2.59)	0.62
Healthy Sleeper			
Frontal	49.01 (8.81)	48.07 (8.73)	0.94
Lateral	50.78 (8.81)	51.72 (8.73)	-0.94
Neck Bent	5.27 (5.51)	5.59 (5.72)	-0.32
Body Bent	2.84 (4.41)	5.28 (5.50)	-2.44
Spine S	0.98 (2.62)	1.73 (3.01)	-0.75
Normal	90.70 (7.47)	87.19 (8.77)	3.51
Movement	7.50 (1.37)	8.07 (1.42)	-0.57
Average of absolute differences			1.26

B. Comparisons of the final simulation results

This work has also assessed the final simulation results including the time percentages of each sleeping pose and the time percentage of movement. The latter one was not validated yet in the step-by-step comparison as it is a global measure. We used the same 100 simulations per sleeper type as in the previous section. Table II shows the average results and the SD between parentheses of the final simulation results for each metric and sleeper type. This table compares the results observed from the sleeper and the ones estimated by presenting the differences of means.

The average of absolute differences of means of all the final simulation metrics and all the sleeper types was 1.26%. It is worth mentioning that the mean differences of time percentage of movement were equals 0.62% or below it for all the sleeper types. This confirms that the simulator also obtained appropriate values for the global movement measure.

Regarding the sleeping poses, the neck bent and spine S poses obtained the lowest mean differences, while the highest ones were obtained by the normal and body bent postures.

In order to assess which of these differences are significant, we conducted several statistical tests. We focused on comparing both the variances and the means. In particular, we applied the Levene's test for assessing the equality of variances. We conducted a t-test for comparing the means as normally done for comparing observed and estimated results of ABSs [38]. Table III shows the results of these two statistical tests. In

TABLE III
RESULTS OF THE LEVENE'S TEST AND THE T-TEST FOR COMPARING FINAL RESULTS ALONGSIDE THE COHEN'S D EFFECT SIZE

	Levene's Test for Equality of Variances		t-test for Equality of Means		Effect size
	F	Sig.	t	Sig. (2-tailed)	Cohen's d
Bad Sleeper					
Frontal	0.852	0.357	2.064	0.040*	0.29
Lateral	0.852	0.357	-2.064	0.040*	-0.29
Neck Bent	0.001	0.890	0.788	0.431	0.11
Body Bent	0.887	0.347	-2.756	0.006*	-0.39
Spine S	0.040	0.841	-0.273	0.785	-0.04
Normal	0.218	0.641	1.485	0.139	0.29
Movement	0.236	0.628	0.627	0.531	0.09
Restless Sleeper					
Frontal	1.440	0.231	2.480	0.014*	0.35
Lateral	1.440	0.231	-2.480	0.014*	-0.35
Neck Bent	0.179	0.673	-0.207	0.836	-0.03
Body Bent	8.393	0.004*	-4.657	0.000*	-0.66
Spine S	4.363	0.038*	-2.480	0.014*	-0.35
Normal	5.034	0.026*	5.076	0.000*	0.72
Movement	2.328	0.129	1.827	0.069	0.26
Healthy Sleeper					
Frontal	0.000	0.986	0.762	0.447	0.11
Lateral	0.000	0.986	-0.762	0.447	-0.11
Neck Bent	0.093	0.761	-0.403	0.687	-0.06
Body Bent	2.977	0.086	-3.465	0.001*	-0.49
Spine S	3.602	0.059	-1.873	0.062	-0.27
Normal	0.929	0.336	3.048	0.003*	0.43
Movement	0.007	0.931	-2.892	0.004*	-0.41

*significant with a .05 level

addition, we calculated the Cohen's d effect size to measure the differences, and the results are included also in table III.

According to the results of the Levene's test, the final simulation outcomes had an equal variance between observed and simulated results for all the simulated metrics in the simulations of the bad sleeper and the healthy one. Nevertheless, the variance was equal in the restless sleeper in four out of seven metrics. The variance of movement time percentage was equals in the three sleeper, which was the metric that could not be assessed in the previous section.

Regarding the significance of the difference of means, four out of seven metrics did not present significant difference between the observed and estimated results in the simulations with the bad sleeper and healthy one. However, in the restless sleeper five out of seven showed significant differences.

Concerning the Cohen's d effect sizes, we interpreted the results following the Cohen's d guidelines [39], which assigned the .2, .5 and .8 values respectively to the small, medium and large effect sizes. In the simulations of the bad sleeper and healthy one, the outcomes only presented small or small-medium effect sizes between the observed and estimated results. However, in the simulation of the restless sleeper, some metrics showed medium-large effect sizes. These medium-large effects were detected when estimating the body bent pose

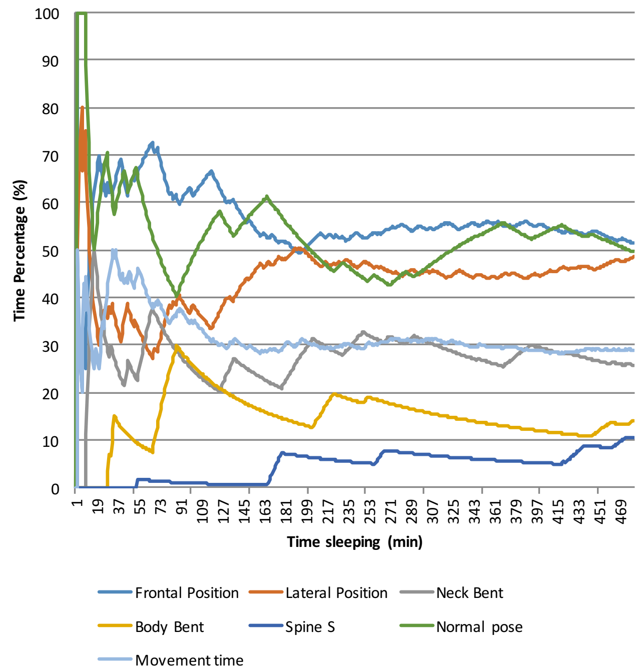

Fig. 8. Example of evolution of a bad sleeper

and the normal one. It is worth highlighting that the movement percentage time had a small or small-medium effect sizes for all the kinds of sleepers.

Therefore, the results advocate that the restless sleeper may be the one that is the most difficult to detect their poses. Its continue changes of poses may make this sleeper adopt stranger combinations of poses difficult to detect. However, the movement metric is properly detected for all the sleeper with small or small-medium effect sizes. The differences of movement were only statistically significant for the healthy sleeper. The reason might be the infrequent movements in this sleeper in comparison to the other two sleeper types. Other works about ABSs like [40] also reported this problem for the estimation of events with infrequent occurrences.

C. Validation of the behaviors of the sleeper types

In order to informally show whether the different sleeper types behaved as expected, we present some examples of their simulated evolutions.

Figures 8 shows an example of a simulation of a bad sleeper. Another example of simulations of bad sleeper was presented when presenting an example of a step-by-step comparison in Figures 6 and 7. Its main feature was that it frequently adopted injuring poses, even if these kinds of poses varied from simulation to simulation. For example, in this evolution, the neck bent evolution values were between 20.2% and 37.9%, and the body bent evolution values were between 7.6% and 29.9%.

Figure 9 presents the evolution of a restless sleeper. Its main feature was the high level of movement activity in comparison to the other kinds of sleepers. This sleeper had a light sleep in almost all the night with a high frequency of movement, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2764467, IEEE Access

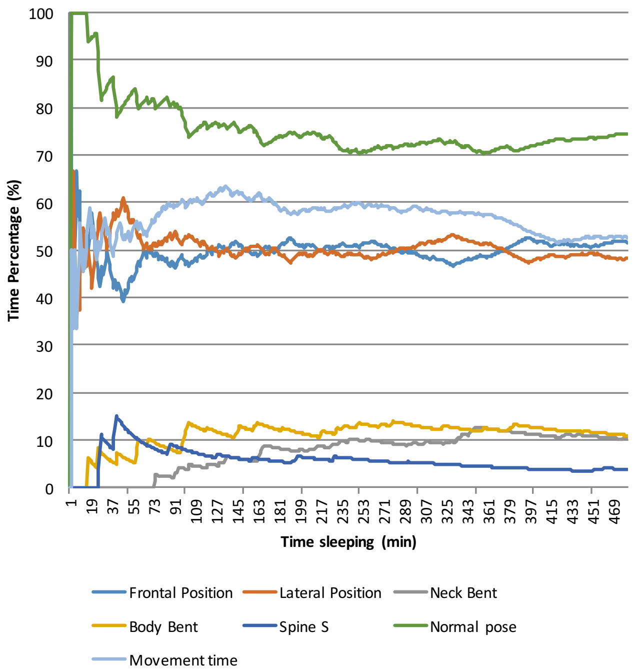

Fig. 9. Example of evolution of a restless sleeper

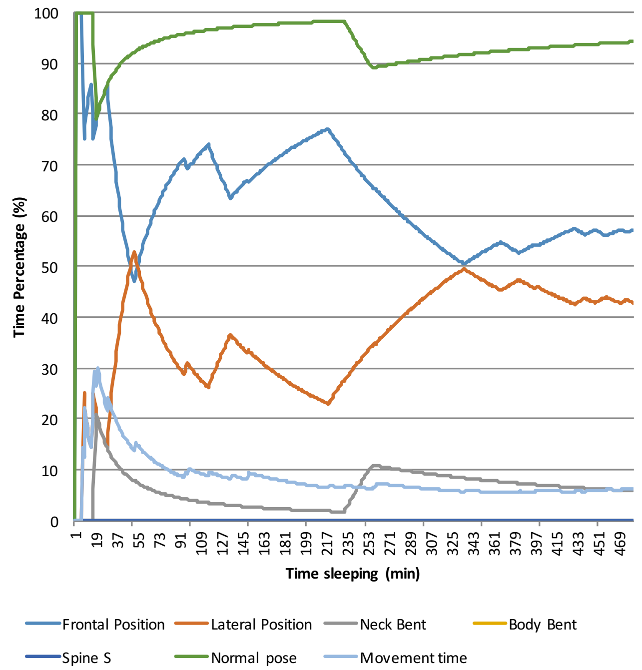

Fig. 10. Example of evolution of a healthy sleeper

this was revealed in the evolution chart. In the last 7 h, the minimum and maximum values of movement time were 51.7% and 63.4% respectively considering one-minute intervals. The average was 57.4 %, and the SD was 5.28%. Due to its high variation of pose rate, the percentage times of injuring poses were more stable in the long-term. This is reflected in the SD of these injuring poses, which were 4.0%, 2.9% and 2.4% respectively for the neck bent, the body bent and the spine S poses.

Figure 10 shows an example of evolution of sleeping poses in a healthy sleeper. It had a low frequency of movements, and most of the time it adopted a normal posture. In particular, the movement time ranged from 5.4% to 13.3%, with an average of 7.0%. The normal pose ranged from 89.1% to 98.3%. with an average of 94.1%. It rarely adopted injuring sleeping poses. In this case, the sleeper never adopted the body bent nor the spine S postures (i.e. maximum value of 0.0%), while it adopted rarely the neck bent posture (i.e. maximum of 10.9% and average of 5.1%). This behavior was the expected one.

Figure 11 shows a visual example of the states of the bed sensors for a bad sleeper after 8 h of sleeping. In particular, the sleeper was in lateral position with the body bent, when the simulation ended.

V. Discussions

The current approach has presented a mechanism for allowing researchers to test different sleeper posture recognition algorithms in a simulated smart bed with IoT and big data generated from their sensors. The underlying framework also allows one to define different sleeper behaviors. This approach has been illustrated by presenting a sleeper posture recognition algorithm. This one obtained a pose detection accuracy higher than the one reported by a similar work of the literature. However, a fairer comparison would require to use a common dataset and the same set of recognized sleeper postures.

For estimating the lateral/front postures, we used two different estimators, respectively based in the shoulders width and the hips width. The former one failed sometimes due to the lateral body bent pose in which the head was detected as part of the shoulders. The estimation through the hips was more accurate, since the head was far from the hips always. However, the variation of the hips is lower between the lateral and front postures. Thus, the estimator of hips needs to be calibrated for the particular sleeper, since this variation may not be perceived if for example their hips widths between lateral and front position are not discriminated by the average threshold. Thus, in the real world, the current work recommends to use the shoulder estimator at the beginning, and offer the possibility of customizing the estimator for a given sleeper with the hips estimator. Moreover, the hips estimator may also need a higher number of sensors per row to notice these small hips variations.

When conducting the experiments, we found that the random change of the sleeper location may be the property that mostly influenced negatively in the ability of the estimator to detect the actual sleeping pose. However, we decided to keep these random movements since the simulations were more realistic, and the variety of cases was much greater. In addition, these variations make room for improving the algorithms for properly detecting sleeping poses from load sensors of smart beds. In this way, in the future, other researchers may use this simulator or the generated logs to test more advanced algorithms in the context of big data analytics.

The current approach allows one to easily test smart beds

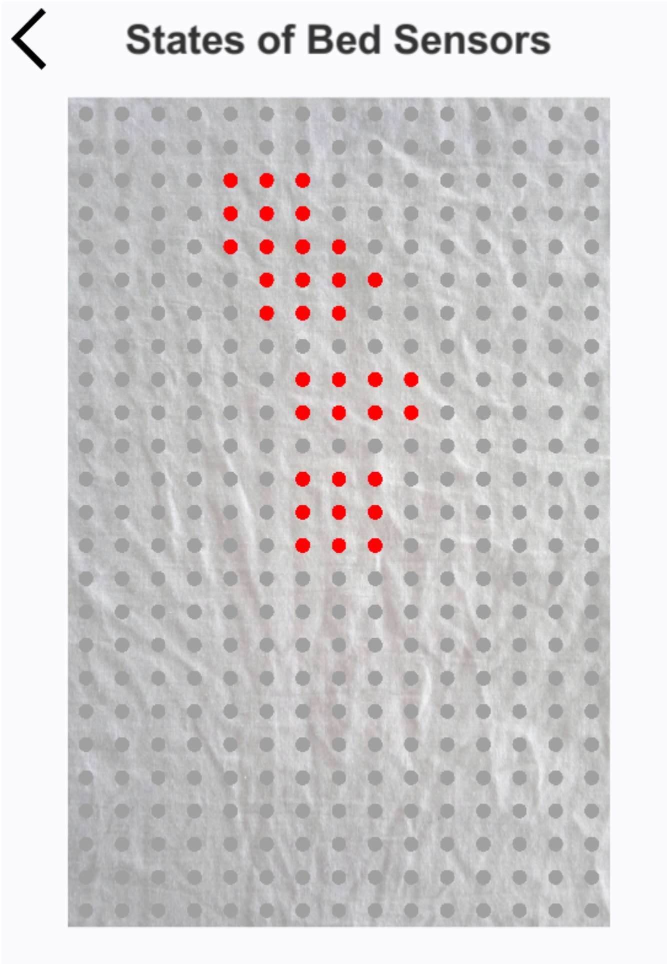

Fig. 11. An example of the states of the bed sensors after a simulation of 8 h of sleeping

with different densities of sensors, using grids with different numbers of rows and columns. This can be useful for estimating which is the minimum amount of sensors to properly detect sleeping postures, in order to be able to build cheap smart beds in the real world. However, the simulator may need to be extended in order to also consider layouts of sensors different from grids, which may obtain better trade-offs between accuracies and costs.

VI. Conclusions and future work

The current work has presented an ABS for supporting the development of algorithms for detecting sleeping postures by simulating a sleeper in a smart bed with IoT and load sensors. The simulator graphically presents the outcomes and allows to compare them with the sleeping postures adopted by the simulated sleeper. It considers both the evolution minute by minute and the final time percentages of each posture. This tool also generates logs for exploring off-line big data analytics. This ABS has allowed us to define an algorithm that recognized properly simulated sleeping postures with a 98% precision, which arguably advocates to outperform the existing algorithm of [10] that reported an 83% precision.

The current work is planned to be extended in several ways in the future. ABS-BedIoT now only visually presents the final pose of the sleeper in the simulation. Since the logs record every pose in the evolution, this can be enough from a big data analytics research viewpoint. However, some end-users may want to observe a visual animation of the transitions of bed sensors. This is planned to be added as an optional feature to the increase the popularity of the application. Moreover, the simulated sleeper movements will be compared with real sleeper movements in order to determine whether the simulator provides realistic simulations of sleepers. Furthermore, the current work is planned to be further assessed by comparing the current algorithm with the one proposed by [10] with a larger dataset that will be used by both of them. In this way, the comparison will be fairer.

Acknowledgment

This work has been supported by the program "Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores" funded by the Spanish Ministry of Education, Culture and Sport with reference CAS17/00005. We also acknowledge support from "Universidad de Zaragoza", "Fundación Bancaria Ibercaja" and "Fundación CAI" in the "Programa Ibercaja-CAI de Estancias de Investigación" with reference IT24/16. This work acknowledges the research project "Desarrollo Colaborativo de Soluciones AAL" with reference TIN2014-57028-R funded by the Spanish Ministry of Economy and Competitiveness. It has also been supported by "Organismo Autónomo Programas Educativos Europeos" with reference 2013-1-CZ1-GRU06-14277. Furthermore, we acknowledge the "Fondo Social Europeo" and the "Departamento de Tecnología y Universidad del Gobierno de Aragón" for their joint support with grant number Ref-T81.

References

[1] H.-J. Yim, D. Seo, H. Jung, M.-K. Back, I. Kim, and K.-C. Lee, "Description and classification for facilitating interoperability of heterogeneous data/events/services in the internet of things," *Neurocomputing*, vol. 256, pp. 13–22, 2017.

[2] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in internet of things," *Computer Networks*, vol. https://doi.org/10.1016/j.comnet.2017.06.013, 2017.

[3] H. Khaleeq, A. Abou-ElNour, and M. Tarique, "A reliable wireless system for water quality monitoring and level control," *Network Protocols and Algorithms*, vol. 8, no. 3, pp. 1–14, 2016.

[4] J. Lloret, E. Macías, A. Suárez, and R. Lacuesta, "Ubiquitous monitoring of electrical household appliances," *Sensors*, vol. 12, no. 11, pp. 15 159–15 191, 2012.

[5] M. Garcia, C. Martinez, J. Tomas, and J. Lloret, "Wireless sensors self-location in an indoor wlan environment," in *International Conference on Sensor Technologies and Applications. SENSORCOMM 2007*. October 14-20, 2007, Valencia, Spain: IEEE, 2007, pp. 146–151.

[6] M. W. Woo, J. Lee, and K. Park, "A reliable iot system for personal healthcare devices," *Future Generation Computer Systems*, vol. https://doi.org/10.1016/j.future.2017.04.004, 2017.

[7] S. K. Sood and I. Mahajan, "Wearable iot sensor based healthcare system for identifying and controlling chikungunya virus," *Computers in Industry*, vol. 91, pp. 33–44, 2017.

[8] R. Mieronkoski, I. Azimi, A. M. Rahmani, R. Aantaa, V. Terävä, P. Liljeberg, and S. Salanterä, "The internet of things for basic nursing care-a scoping review," *International Journal of Nursing Studies*, p. https://doi.org/10.1016/j.ijnurstu.2017.01.009, 2017.

[9] M. Z. Hossin, "From habitual sleep hours to morbidity and mortality: existing evidence, potential mechanisms, and future agenda," *Sleep Health: Journal of the National Sleep Foundation*, vol. 2, no. 2, pp. 146–153, 2016.

[10] J. J. Liu, W. Xu, M.-C. Huang, N. Alshurafa, M. Sarrafzadeh, N. Raut, and B. Yadegar, "Sleep posture analysis using a dense pressure sensitive bedsheet," *Pervasive and Mobile Computing*, vol. 10, pp. 34–50, 2014.

[11] T.-S. Wong, S.-K. Wong, and L. W.-c. Leung, "Sleep-posture sensing and monitoring system," Aug. 12 2014, uS Patent 8,803,682.

[12] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, and Z. Wang, "Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2028–2036, 2010.

[13] R. Fuentes-Fernández, I. García-Magariño, J. J. Gómez-Sanz, and J. Pavón, "Integration of web services in an agent-oriented methodology," *International Transactions on Systems Science and Applications*, vol. 3, pp. 145–161, 2007.

[14] I. García-Magariño and I. Plaza, "ABS-MindHeart: An agent based simulator of the influence of mindfulness programs on heart rate variability," *Journal of Computational Science*, vol. 19, pp. 11–20, 2017.

[15] I. García-Magariño and I. Plaza, "ABSEM: An agent-based simulator of emotions in mindfulness programs," *Expert Systems with Applications*, vol. 84, pp. 49–57, 2017.

[16] A. Rghioui, S. Sendra, J. Lloret, and A. Oumnad, "Internet of things for measuring human activities in ambient assisted living and e-health," *Network Protocols and Algorithms*, vol. 8, no. 3, pp. 15–28, 2016.

[17] J. Lloret, A. Canovas, S. Sendra, and L. Parra, "A smart communication architecture for ambient assisted living," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 26–33, 2015.

[18] K. Liu, C. Wang, and S. Liu, "A novel mobile data collection algorithm for wireless sensor networks," *Adhoc & Sensor Wireless Networks*, vol. 36, no. 1-4, pp. 285–311, 2017.

[19] B. Placzek, "Suppression-based data collection approach for navigation of mobile node in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 35, no. 3-4, pp. 173–192, 2017.

[20] S. Kim, "Game based certificate revocation algorithm for internet of things security problems," *Adhoc & Sensor Wireless Networks*, vol. 32, no. 3-4, pp. 319–336, 2016.

[21] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[22] G. Yang, L. Xie, M. Mäntysalo, X. Zhou, Z. Pang, L. Da Xu, S. Kao-Walter, Q. Chen, and L.-R. Zheng, "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE transactions on industrial informatics*, vol. 10, no. 4, pp. 2180–2191, 2014.

[23] W. SpillmanJr, M. Mayer, J. Bennett, J. Gong, K. Meissner, B. Davis, R. Claus, A. MuelenaerJr, and X. Xu, "A smartbed for non-intrusive monitoring of patient physiological factors," *Measurement Science and Technology*, vol. 15, no. 8, p. 1614, 2004.

[24] A. Hart, K. Tallevi, D. Wickland, R. E. Kearney, and J. A. Cafazzo, "A contact-free respiration monitor for smart bed and ambulatory monitoring applications," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 927–930.

[25] S. H. Park, S. H. Won, J. B. Lee, and S. W. Kim, "Smart home–digitally engineered domestic life," *Personal and Ubiquitous Computing*, vol. 7, no. 3-4, pp. 189–196, 2003.

[26] S. H. Ahmed and D. Kim, "Named data networking-based smart home," *ICT Express*, vol. 2, no. 3, pp. 130–134, 2016.

[27] K. Xu, X. Wang, W. Wei, H. Song, and B. Mao, "Toward software defined smart home," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 116–122, 2016.

[28] M. Rooney, P. Fess, D. James, J. Myers, J. Murphy, J. Taylor, and L. Torii, "Effect of mattresses and pillow designs on promoting sleep quality, spinal alignment and pain reduction in adults: Systematic reviews of controlled trials," Ph.D. dissertation, Utica College, 2015.

[29] C. L. Adler and Y. R. Zarchin, "The virtual focus group: Using the internet to reach pregnant women on home bed rest," *Journal of Obstetric, Gynecologic, & Neonatal Nursing*, vol. 31, no. 4, pp. 418–427, 2002.

[30] I. García-Magariño, A. Gómez-Rodríguez, J. C. González-Moreno, and G. Palacios-Navarro, "PEABS: a process for developing efficient agent-based simulators," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 104–112, 2015.

[31] I. García-Magariño, M. Cossentino, and V. Seidita, "A metrics suite for evaluating agent-oriented architectures," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 912–919.

[32] J. Bravo and I. García-Magariño, "A Methodology for elaborating activities for higher education in 3D virtual worlds," *Journal of Interactive Learning Research*, vol. 26, no. 1, pp. 23–38, 2015.

[33] I. García-Magariño, C. Gutiérrez, and R. Fuentes-Fernández, "The ingenias development kit: a practical application for crisis-management," *Bio-Inspired Systems: Computational and Ambient Intelligence*, pp. 537–544, 2009.

[34] B. Klefbeck, M. Sternhag, J. Weinberg, R. Levi, C. Hultling, and J. Borg, "Obstructive sleep apneas in relation to severity of cervical spinal cord injury," *Spinal cord*, vol. 36, no. 9, pp. 621–628, 1998.

[35] S. J. Gordon, K. A. Grimmer-Somers, and P. H. Trott, "A randomized, comparative trial: does pillow type alter cervico-thoracic spinal posture when side lying?" *Journal of multidisciplinary healthcare*, vol. 4, p. 321, 2011.

[36] I. García-Magariño, G. Palacios-Navarro, and R. Lacuesta, "TABSAOND: A technique for developing agent-based simulation apps and online tools with nondeterministic decisions," *Simulation Modelling Practice and Theory*, vol. 77, pp. 84–107, 2017.

[37] S. Brand, M. Gerber, N. Kalak, R. Kirov, S. Lemola, P. J. Clough, U. Pühse, and E. Holsboer-Trachsler, "Adolescents with greater mental toughness show higher sleep efficiency, more deep sleep and fewer awakenings after sleep onset," *Journal of adolescent health*, vol. 54, no. 1, pp. 109–113, 2014.

[38] T. Filatova, D. Parker, and A. Van der Veen, "Agent-based urban land markets: agent's pricing behavior, land prices and urban land use change," *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 1, p. 3, 2009.

[39] J. Cohen, *Statistical power analysis for the behavioral sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Earlbaum Associates, 1988.

[40] I. García-Magariño, C. Medrano, A. S. Lombas, and A. Barrasa, "A hybrid approach with agent-based simulation and clustering for sociograms," *Information Sciences*, vol. 345, pp. 81–95, 2016.