Research Article

An IoT Platform for Epilepsy Monitoring and Supervising

Paula M. Vergara,[1] Enrique de la Cal,[1] José R. Villar,[1] Víctor M. González,[2] and Javier Sedano[3]

[1]*Computer Science Department, University of Oviedo, Campus de Viesques, s/n, 33204 Gijón, Spain*
[2]*DIEECS, University of Oviedo, Campus de Viesques, s/n, 33204 Gijón, Spain*
[3]*Instituto Tecnológico de Castilla y León, Polg. Ind. Villalonquejar, López Bravo 70, 09001 Burgos, Spain*

Correspondence should be addressed to José R. Villar; villarjose@uniovi.es

Epilepsy is a chronic neurological disorder with several different types of seizures, some of them characterized by involuntary recurrent convulsions, which have a great impact on the everyday life of the patients. Several solutions have been proposed in the literature to detect this type of seizures and to monitor the patient; however, these approaches lack in ergonomic issues and in the suitable integration with the health system. This research makes an in-depth analysis of the main factors that an epileptic detection and monitoring tool should accomplish. Furthermore, we introduce the architecture for a specific epilepsy detection and monitoring platform, fulfilling these factors. Special attention has been given to the part of the system the patient should wear, providing details of this part of the platform. Finally, a partial implementation has been deployed and several tests have been proposed and carried out in order to make some design decisions.

1. Introduction

Epilepsy is a chronic neurological disorder characterized by involuntary recurrent convulsions [1]. There are about 65 million people affected all around the world, with a high and dramatic impact not only on the patient's quality of life, but also on the professional development and social behaviour; the health system budget is highly affected as well.

The illness anamnesis improves with the existing platforms for patient monitoring and weblogs. The main part of these platforms has been developed for two different and most frequent kinds of epilepsy crisis: the generalized tonic-clonic seizures and the typical absence seizures [2]. In these two cases, the detection of a seizure can be efficiently faced using wearable sensors (WDs) including a triaxial accelerometer (ACM) and/or a heart rate (HR) sensor: the former type detection has been reported in [3], the latter one has been characterized in [4], and an HR-based detection system has been proposed in [5].

Another main aspect of the anamnesis process is where the data is gathered. The main part of the literature deals with constrained spaces, that is, research laboratories or hospital rooms [6], or even the patient's house [7], but without considering the normal everyday life [8, 9]. We claim that the data should be gathered in everyday life, allowing the patient to freely decide what to do and how to do it. This is important because, firstly, the data is gathered from normal activities performed before and after a seizure, and secondly, the analysis and procedures should adapt to this unconstrained world, making the whole detection process much more difficult.

A careful in-depth analysis of the seminal papers concerning epilepsy monitoring platforms [10] and Mobile Cloud Computing (MCC) [11–13] let us conclude that the current available platform, either in the scientific literature or in the market, lacks several main features that are not comprehensibly integrated. These include, among others, the instantaneous data source problem, the ergonomic aspects in the design, the deployment cost of the solution, and the energy efficiency of the approach.

This study aims to solve some of these limitations; to do so, a solution is proposed and an experimentation stage has been performed in order to extract the suitable conclusions for the epilepsy monitoring platforms. In the next section, the

most relevant contributions in the literature are analyzed and criticized, paying special attention to the published platforms; the main concerns that remain unsolved are included as well. Section 3 is devoted to explaining an architecture that solves the main concerns found in the literature related to the design of epilepsy monitoring platforms. This section also describes the MCC and discusses some design parameters. Finally, an experimentation stage has been performed and the results are included and discussed in Section 4, which allows us to produce the final conclusions. For the sake of readability, Abbreviations includes the most relevant acronyms used within the text.

2. Related Work

Several different eHealth platforms have been released and reported in the literature for the detection and/or monitoring of illnesses in real time [14], even using WD and/or Body Sensor Networks (BSN) [15, 16]. Typically, the WD are proposed for data gathering, measuring biomedical variables, or obtaining feedbacks from the users, either performing local MCC processing [17, 18] or requesting Cloud Computing (CC) services [19–21]. Usually, the CC services are responsible for processing and storing the sampled data from sensors, as well as the model learning, and those computational greedy tasks. Additionally, the CC services also provide the presentation layer, in terms of user alarms to the patients or medical staff, notifications to the patient's relatives, or even performing graphics and data analytics for further studies.

Examples of such platforms include CoCaMaal, ROCHAS, and AACMPE. CoCaMaal, the acronym for A cloud-oriented context-aware middleware in ambient assisted living [21], is specialized in the patient monitoring and in the event control, either notifications or accidents. This platform restricts itself to fully controlled environments as long as it suggests the deployment of BSN according to the patient's conditions. Variables such as electroencephalogram (EEG), electrocardiogram (ECG), ACM, HR, and blood pressure are considered to be placed on each patient; therefore, the ergonomic aspects of this solution need checking.

A second interesting platform is ROCHAS (Robotics and Cloud-assisted Healthcare System for Empty Nester) [22], which proposes the monitoring of handicapped patients in their own home, allowing them to live as independently as possible by means of an assistance robot. Similarly, an assistant platform for elderly people was proposed in a series of studies in [19, 20, 23], where open software platforms are analyzed to work together.

AACMPE, short term for Allergy and Asthma Care in the Mobile Phone Era [24], controls the allergies and asthma evolution of the patients using MCC to determine several variables: the peak exhalation flow and the peak nasal inhalation flow and some breath parameters and sounds, among others. The Chinese CMTHC project, Children's medical treatment and healthcare system [25], has been reported to monitor unhealthy children by means of web logs to be completed with sensor data, like body temperature, HR, and so on. A further step has been proposed in AIWAC, affective interaction through wearable computing and cloud

technology [26], which analyzes the affective needs of the patients based on the measurements obtained from a WD. Other platforms that propose open solutions for monitoring illnesses introduce BSN in a more abstract way. Further work on adapting them for epilepsy detection is needed [27, 28].

Concerning the epilepsy detection and monitoring, several approaches have been reported in the literature. The main data source to do so is the EEG, measuring the electrical activity of the brain to detect the epileptic seizures. Advances in some issues have been published, like modelling the recorded signals [29, 30] or the design of portable EEG devices to deploy such models.

Some examples of these epilepsy specific platforms include EpiCare, Sareen, or Bajwa. In EpiCare, a home care platform based on Mobile Cloud Computing to assist epilepsy diagnosis [7], portable EEG are used to detect the seizures in controlled environments. Sareen gathers data from EEC, using MCC and CC for storage and for sending notifications to the relatives and medical staff, including location information [31, 32]. The main difference between Sareen and Bajwa is that the latter proposes CC only.

EEG aside, different biometric signals have been proposed for detecting the epileptic seizures [10, 33]: gyroscopes, magnetometers, implanted advisory system, electromyography, ECG, ACM, video detection systems, mattress sensor, and audio classification. BSN placed on the body together with MCC has been proposed for epilepsy detection in [8]; more specifically, the BSN includes an ACM cap, a wrist band including a temperature sensor and an ACM, a moisture sensor, and microphone. The MCC layer analyzes the data and detects the seizures; the relevant data is stored and transmitted to the medical staff for further analysis.

Furthermore, solutions making use of CC services have been also reported, mainly for the storage and modelling of the gathered data. For instance, not all the platforms store the data stream; on the contrary, the majority of them process the data as it comes in order to generate the alarms and delete them afterwards: these solutions rely on previous research stages that would have been performed to obtain the deployed models [6, 8, 9, 34–38]. On the other hand, there are solutions where the gathered data is stored for data analytics [39, 40] or even for future use [41–44]. Finally, complete CC solutions have been also tackled in the literature, including not only the data storage and visualization but also the modelling and classification of the current state of affairs [7, 31, 32, 45–48].

Soon after the study of Schulc et al. [36], ACM was proposed for epileptic seizure detection; since then, plenty of studies concerning with the detection of this type of seizure have been published, focusing only on the machine learning issues. For instance, a wrist band including an ACM connected to a Smartphone was proposed for MCC-based seizure detection [3, 6, 49]; no further connection with CC services was considered. ACM, gyroscopes, and magnetometers were proposed in [9] as the BSN, taking advantage of the local Smartphone for analyzing the data and sending e-mails to the medical staff with the patient position. Similar works were presented in [47, 48]. LabView has been used for developing a solution as well [35] by using an Arduino ACM sensor to generate alarms that are transmitted to the medical service.

SmartMonitor® devices have been used for detection of limb shaking, sending the alarms to a website linking the WD with a Smartphone [44]. ECG for detecting epileptic seizures while sleeping has been reported in [39], while EEG hats have been also effectively used in seizure detection [41, 50]; however their ergonomic characteristics make them difficult to use. Further studies in this context of epileptic seizure detection include BSN sending information to a computer in controlled environments [42, 43]; ACM, temperature, and skin humidity data gathered to computers [40]; ACM and HR [5]; or the use of thresholds as a detection method when measuring the HR linked to a Smartphone [38].

Finally, plenty of apps have been published in the corresponding markets for the detection using either the Smartphone sensors or external sensors, EpDetec [51] and MyEpiPal [34], or for the web logging, facilitating the way a patient records daily information concerning her/his epileptic events, medication, and news, My Epilepsy Diary [52], EpiDiary [53], Epilepsy Society [37], and Epi & Me [54]. A reader who is interested should refer to [55] for a performance comparison of epilepsy related apps.

2.1. Remarkable Factors. After this thorough analysis of the literature, we have found out that several remarkable factors have not received enough attention from the research community concerning epilepsy seizure detection and patient monitoring platforms.

Real Time Response. The detection of seizures and the alarm notification should work almost immediately. That means that once a seizure sets in, in the smallest time lapse the alarms should notify the relatives and the health services. A balance between stand-alone solutions, that is, everything computed in the mobile device, and the autonomy and the battery life should be achieved: the higher the computation, the shorter the battery life and thus the autonomy. Therefore, some services must be declared as essential, while others might be postponed, storing the information in a local database. Furthermore, the features and capabilities for each mobile device must be properly introduced in the system, so a suitable ordering of the services can be generated.

Ergonomic Issues. The solutions should be easy to wear and unobtrusive: the better the wearable conditions, the smaller the chances of finding a suitable excuse for not using the solution. Ergonomic issues also include factors like energy efficiency and long battery life: the lack of them forces frequent charging cycles of the devices and some other annoyances. Besides, some examples of devices without lacking in ergonomic issues are found in the main part of the solutions; they are efficient for their purpose, but their use is uncomfortable though: using sensing caps [7, 8, 31, 32], wrongly sized WD [36, 41], too many WDs [8], and so on.

Deployment Cost of the Platform. As for the ergonomic issues, no further study has been considered about the cost of the solution: are the WDs affordable? How much computation would be needed? Can the solution be delivered as a general service? Are tuning stages required? Is specific training of the users needed? Are the solutions economically feasible? How are they contributing to the health system and at what cost? Unfortunately, these questions have been sparsely answered in the published solutions.

CC Guaranteeing the Service Delivery. CC services should be responsible for making all the services available to the users. In spite of the distribution of the computational tasks among all the available hardware, this CC layer should solve any task that no other element within the infrastructure can afford. Besides, complex tasks, such as extension mechanisms that include on-demand model learning, need to be addressed in this layer as well: learning algorithms need relatively large datasets and high computational resources that by no means can be faced on, for instance, mobile devices without penalizing the battery life. Therefore, according to the current scenario and user, there would be the need to store data from the sensors within the CC layer. However, there could be some learning tasks that can be distributed as well, such as those related to active learning stages that might be employed.

All of these considerations lead back to web service ontology development [56–58], where the different services, and tasks, were completely defined as well as the relationships among them. This knowledge representation allows a mediator, or scheduler, to set out a plan and allocate the tasks and services on the different nodes. Extending such ontologies with the actuators, the set of computational nodes and devices, and their features would allow scheduling the tasks for each scenario. Furthermore, new services, and tasks, that extend the functionality of the platform can also be described in ontology terms, so their use can be hot-deployed without stopping and reconfiguring the platform.

MCC Integration. When required, the Smartphone can take the control of the sequence of tasks to perform. This would happen mainly when no Wi-Fi network is available for sending the data; actually, this MCC might help in balancing the battery life and the networking load, reducing the required 4G data limit and, thus, the fares to pay. With MCC we understand a combination of CC services together with mobile computing interconnected by means of wireless networks [59–63]. However, we move one step forward in the mobile computing part, introducing the concepts suggested in [12, 64]: we propose the use of hybrid mobile applications, where several services provided by the CC can also be dynamically allocated and performed by the mobile computation, provided there are resources available. In other words, the mobile device can perform as a local cloudlet.

Configurable Services within the Apps. Instead of running all the possible stack of services, it should be desirable that the app adapts to the patient as much as possible, disabling those services that are no longer needed while, at the same time, including new added value services, such as web logging capabilities and affective computing issues. Again, ontologies can help in deploying these issues.

A comparison between the different mentioned platforms solutions is shown Table 1, paying attention to those factors that have been considered relevant within the

TABLE 1: Main epilepsy detection platforms and solutions published so far. A description with some concerns with respect to the relevant factors is detailed. We suggest [10, 55] for further reading.

Ref.	Description
[41]	An EEG sensory device linked to a Smartphone, which performs some processing of the data, is used together with CC services. However, this is a very short communication and does not provide details of the solution.
[6]	This is mostly a research study focused only on the detection of tonic-clonic epileptic seizures. A WD without wireless communication stored data and some machine learning methods were performed for obtaining offline models.
[7]	In EpiCare, the Android app runs on the Smartphone device, gathering data from a EEG electrodes cap. This hardware makes this solution very efficient but rather uncomfortable. The main thing is that the project is focused on SUDEP; thus it is feasible to have a cap while sleeping. A mixture of CC and MCC solution is proposed, suggesting that an intelligent task delivering and allocation should be performed but without proposing any viable technique.
[34]	MyEpiPal is an app that monitors the patient, simplifies the intercommunication between the caregiver and the patient, and allows the self-management. This means that although it makes use of the sensory within the Smartphone, the main goal is to give support to the patient in everyday life. It is not an epilepsy detection platform itself, although the measurements can help in the prediction of the occurrence of a seizure, which is the reason that it is included in this comparison.
[39]	This research details the design of an ad hoc epilepsy detection ECG wireless intelligent sensor, including several detection algorithms, linked to a local computer connected to a network. Several relevant factors were analyzed, the ergonomic issues and the battery life among them. The WD communicates with the local computer in order to deliver alarms, to receive configuration commands, or to start/stop HR recording to be downloaded to the computer. A very detailed explanation of the requirements and of the hardware decisions is included.
[31, 32]	An EEG cap linked to a Smartphone is arranged to send the gathered data to CC services. Whenever a seizure is detected on the cloud, GPS locations are shared through the notification system. Neither the ergonomic issues nor the battery life and autonomy and the economic costs of the platform have been analyzed.
[37]	The UK Epilepsy Society published an app for Android and iPhone as a Web-log of the seizures, medication monitoring, and so on. This solution is a standalone solution, and the data gathered with this tool is not shared with any health service.
[38]	A standalone mobile solution is proposed using a commercial Smartwatch (MIO Alpha) linked to an Android phone. The app is responsible for gathering and processing the data, as well as the modules for the epileptic seizure detection. The solution does not consider intelligent modules, just some thresholds; this technique is by no means valid in this type of detection.

literature: deployment cost of the platform, real time response, ergonomic issues, MCC integration, multiple services within the Apps, and, finally, CC online modelling.

As can be seen, the problem of designing a platform for the specific problem of epileptic seizure detection and monitoring has not been completely addressed in the literature. Although some general design principles are still valid, some of them need special care to propose a final solution. The next section deals with the decisions for the design of an epileptic monitoring and supervising platform, providing an abstract architecture and the description of a prototype for the evaluation of some of the above-mentioned factors. These decisions include ontology driven tasks, dynamic data gathering and modelling, and the integration of MCC and CC.

3. A Platform for Monitoring and Supervising Epileptic Patients

This study proposes a platform for monitoring and supervising epileptic patients, focused on the two main epilepsy types: the focal myoclonic and the epileptic absence seizures. This platform aims to solve each of the most relevant factors seen in the previous section, providing some extensibility to

future developments. This section is devoted to describing this platform as well as the developed prototype. The next subsection introduces some considered concept decisions and requirements, while Section 3.2 describes the abstract architecture. The remaining part of the section, Section 3.3, focuses on detailing the developed prototype.

3.1. Design Decisions and Requirements. This study proposes the use of noninvasive WD such as a sensory bracelet plus a Smartphone to allow the patients to carry on with their own life, performing their everyday activities without the need for specific clothes or garments. In this study, a pair of a WD linked by means of bluetooth 4.0 Low Energy to a Smartphone is called a patient's kit (PK). This solution enhances the ergonomic issues of the solution while favouring the patients to continue using the system. The WD should include ACM and HR sensors in order to detect the two focused types of epileptic seizures. As mentioned in the related work section, there are several studies for detecting the focal myoclonic type of seizure, while further study is needed to detect the second focused type. In order to detect the focal myoclonic seizures, the research published in [3, 65] is proposed. Further study is needed to solve the second type of seizures. The selection of those techniques has been taken due to two main

reasons: (i) the obtained results and (ii) the simplicity of the used models. This simplicity would eventually allow them to be implement so they can run on any available platform, the Smartphones among them.

Taking advantage of the computational capacities of mid-range Smartphones, this study proposes that, besides data gathering and processing and perhaps simple thresholding, the MCC services can be extended to incorporate local model evaluation. To do so, incremental deployment/delivery of trained/tuned models into the MCC kernel would allow continuing the monitoring, providing real time response even when the CC is totally unavailable through Wi-Fi networks. Nevertheless, with the aim of extending the battery life as much as possible, a mixture of MCC and CC together with a suitable balancing algorithm to decide where the decisions or calculations are shall be accomplished. The study of balancing MCC computation, decreasing the communication acts, and the CC computation, decreasing the computation amount in the Smartphone, is one of the contributions of this work, as will be shown later.

Besides, ontology driven tasks and dynamic data gathering personalization are required in order to extend the system. With ontology driven tasks we refer to designing and developing ontologies that describe every single task: from sampling to alarming and notifying. The concept of ontology driven tasks facilitates extending the system, so new procedures can be easily conceptually developed and distributed and deployed in any of the available computation layers. On the other hand, dynamic data gathering personalization refers to marks for which patients data should be gathered. For instance, developing new services, say, detecting a new type of seizure, needs specific data to be gathered for further processing and analysis. However, it is impossible to gather data from every patient as long as the amount of data to store grows unbounded. In addition, it is better to gather data only for patients that might perform the desired event to detect or identify. Actually, the development of ontology driven tasks allows the dynamic data gathering to be implemented as the latter can be viewed as a new task devoted for a specific group of patients.

In addition, the CC services should be performed on low-cost servers that can even be deployed in different public sites, in outpatient clinics, for instance, shifting the computation resources to the endpoints. These servers on the far edge must be federated to avoid data losses and to enhance the performance of the whole system. Furthermore, local nodes with available unused computing resources, personal computers or even personal servers, can be designated to become part of the solution, enhancing the overall computational capacities of each installation while keeping the low-cost profile. Clearly, for this latter case, it might be advisable to sign a special commitment between the user and the health system for accomplishing data and privacy regulations.

Finally, some decisions should be taken regarding with the data analysis and data monitoring and with the system extension capabilities. On the one hand, the Exploiting and Data Analytics module is needed. This module tackles the monitoring and tracking of the patients, showing the main facts to the medical staff. This module also includes human-machine interfaces; the main part of these interfaces should be light clients, HTML5/REST clients or similar, based on Bootstrap technology, so they keep the responsiveness. On the other hand, an extension mechanism must be provided. For instance, the data analysis should allow the medical unit staff to perform a high level of machine learning experimentation that might lead to models for detecting or enhancing different types of epileptic seizures. Intelligent interfaces, similar to KEEL [66] or WEKA [67], should define the offline tasks, their outcomes, and reports. In addition, these interfaces should also allow defining new sequences in the ontology driven system outlined before. For sure, unless the medical unit staffs incorporate multidisciplinary teams, these extension mechanisms must be kept simple and elementary.

3.2. The System Abstract Architecture. Let us call a scenario a concrete specification of the computing devices that are available for solving all the calculations needed to detect and monitor a certain patient. Let us assume that a complete ontology of services, tasks, computing devices, and scenarios is obtained; this ontology could be based on that presented in [56]. Let us assume as well that this ontology is fulfilled for the algorithms, tasks, and services involved in this project. Let us also assume that the algorithms and tasks have been properly implemented to run either on an app, on a server, or on both and that all these varieties are reflected in the ontology data.

According to [56], any sequence of tasks can be completely represented in the ontology, and a mediator can locate each of them on a concrete machine. In this approach, we assume a mediator that assigns the tasks to be executed and where they are to be carried out for each case. Therefore, for each patient and scenario, a sequence of tasks can be planned and allocated. In other words, we are able to define a specific planification and task allocation for each patient and scenario; both of them are made explicit in ontology basis; and every single computing device, including the Smartphones, has access to this information.

The proposed abstract architecture is depicted in Figure 1 as a solution for this very specific epilepsy monitoring and supervising platform. A PK is conformed with a WD plus a Smartphone; the Smartphone includes a complete app that, together with the ontology and the current scheduling, performs the sensor sampling, and so on. Whenever available, Wi-Fi networks are used to send data bunches to be stored in the health service. Nevertheless, notifications and alarms, when generated, should be delivered using the available connectivity. In addition, some spaces can host specific hardware performing as a federated CC server. In any case in which Wi-Fi connectivity is available, the Smartphone must delegate on these systems in order to alleviate the computational requirements for the sake of extending the battery life.

Besides, the health system services, both the CC services and the data storage, perform all the data storage and the computation that remain unsolved in the system, including detection of seizures, alarm generation, and notifications, as well as the reporting for the medical staff. The central CC services, those reflected in the DMZ zone, cope with
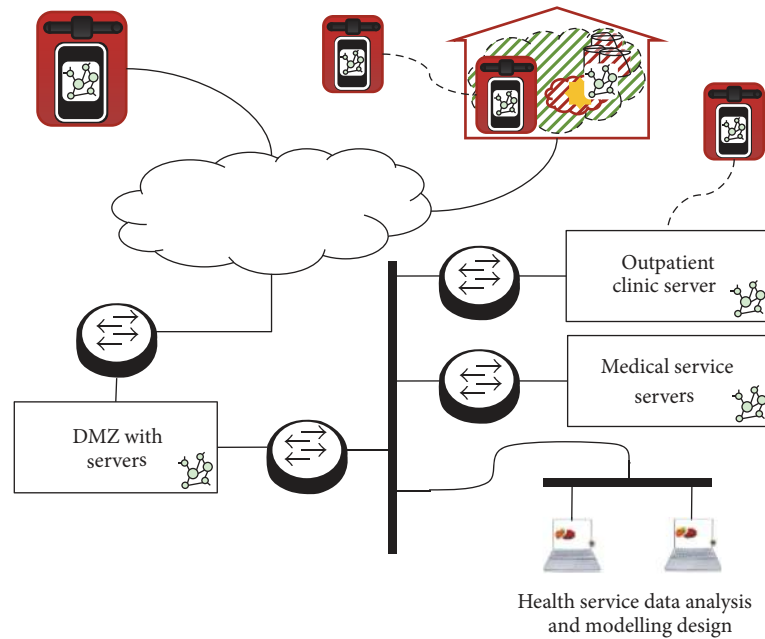
Figure 1: The architecture proposed in this study. Every single computing element has access to the ontology, the patient's tasks planning, and allocation for the current scenario. The PK can work alone or using the available CC services. Whenever possible, the configuration that leads to longer battery life will be chosen. CC services on the health service or in the specific medical unit service perform the calculations and data storage. Data visualization and report generation are typical in the data exploitation. Moreover, the medical unit should be able to perform data analysis and modelling using intelligent interfaces that integrate the needed ontology information in order to distribute the tasks.

those services that are required for a suitable performance of the epilepsy seizure detection and patient monitoring, while those services and tasks devoted to extract new knowledge should be carried out on specific servers belonging to the corresponding medical unit.

This abstract definition of the architecture needs completion: the ontology development and fulfillment, the infrastructure, the mediator implementation, and so on; however, it cannot be successfully detailed in a single study due to its own complexity. The remainder of this study focuses on a prototype developed that includes a MCC solution and the PK because it is the minimum part required that allows us to evaluate some parameters of the system. For this prototype, neither the ontology nor the extending capabilities have been introduced. However, this data gathering prototype is needed in order to obtain real data from epileptic patients, allowing us to develop the remaining modules.

3.3. The MCC and Monitoring Unit. As mentioned before, the PK includes a WD and a Smartphone. The sampling frequency of the sensory system should depend on the physical measurements: accelerometers need sampling frequencies higher than 10 Hz [3]; HR needs smaller sampling frequencies. However, the majority of the commercial Smartwatches or Smartbands do not allow apps to sample data from the sensory system: they only allow access to calculated transformations or induced variables. The WD manufacturers offer their own SDK, which may or may not allow reading the raw data from the sensor; in the majority of the cases, the sampling frequency might not be fixed.

The majority of the products give access to a website where the aggregate variables are available, for instance, the burnt calories, but they do not store the instantaneous data. Besides, the main part of the HR sensors allows downloading the data but not by streaming, disabling the online process of the signal. Further requisites for the WD include the use of low energy consumption networking, such as bluetooth 4.0 Low Energy, and a valid battery duty cycle of about one day long. To our knowledge, the single marketed solution that was valid for this type of applications was Pebble; unfortunately, the company was acquired and closed by a competitor. Recently, Samsung's Gear 2 devices got their market price decreased, allowing them to be considered as valid candidates; nevertheless, this fact has happened while writing this study, so they have not been tested yet. Therefore, currently available solutions are not suitable for the PK; thus, in this study we proposed an ad hoc solution including 3DACM and HR; interested readers can get more information concerning this WD in Section 4.1.

The structure of this current proposal for MCC is mainly based on the challenges described in [13] for the definition of MCCs. The costs subsection is inspired by that of cost analysis detailed in [12]. Figure 2 shows both the scheme of the PK and the MCC architecture. The architecture design decision needs further detail; the next subsection gives details concerning this MCC layer. From Section 3.3.2 to Section 3.3.4, several design parameters are analyzed: the data partitioning reliability, the platform's privacy, the network and the energy efficiency, and the machine learning issues, correspondingly.
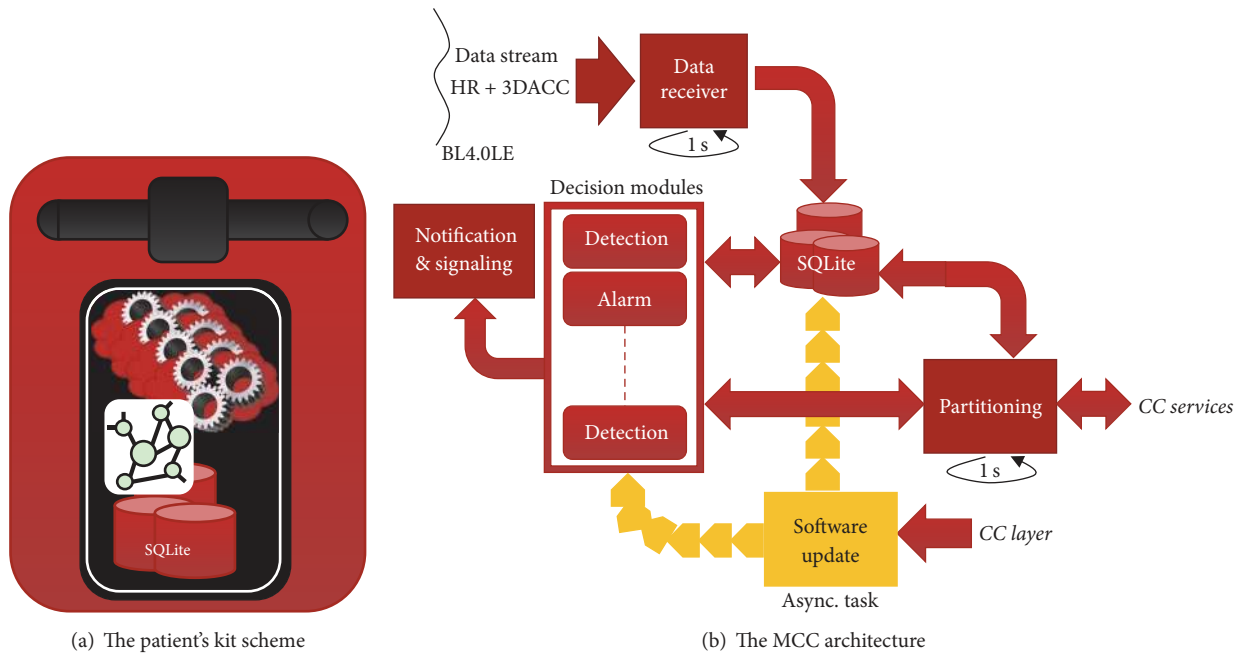
FIGURE 2: (a) The PK scheme. (b) The MCC architecture. In this study Android Smartphones have been used; therefore the SQLite database management system is used. The different services refer to the capacity of the software (i) to receive data from the WD and to send data to the CC layer, (ii) to receive and send notifications and alarms, (iii) to process the received data from the WD, and so on. The 1-second loops refer to tasks that are scheduled periodically. The software update also induces changes in the database; for instance, a Decision Tree classifier can retrieve its rule from the database. The changes in the software refer, then, to the improvements in the algorithm implementation to enhance the energy efficiency.

3.3.1. The MCC Architecture. Several different approaches for MCC architectures have been described in the literature, from the most centralized approaches, focused on very light clients and a powerful cloud part, like in the Centralized Cloud [11, 68], to the hyperdistributed and high network synchronization requirements, like cloudlet [69]. For the purpose of this study, we claim that an ad hoc decentralized solution (Mobile Ad Hoc NETworks, MANET [70]) is the most suitable architecture.

The MANET architecture provides a high level of autonomy, with the local storage capacity and data processing. Moreover, requests to heavy and computational expensive cloud services are allowed as well. Besides, the cloudlet solution is not suitable because there is a single wearable which might be connected to the MCC layer and, mainly, the data from each patient are totally independent of one another; this latter fact suggests that there is no need for MCC synchronization among MCC layers, relaxing the computational requirements of this layer.

Conversely, our proposal manages the Smartphone as a service node, being responsible for storing the instantaneous data and its transformations, performing low-medium computation tasks, and so on. Therefore, this solution is highly based on that of MANET Mobicloud [70], allowing distributed and collaborative CC services to offer their solutions when the network is available [71, 72].

Figure 2(b) shows the proposed architecture in detail. Two main tasks, the Data Receiver and the Partitioning tasks, are continuously dispatched by means of a timer. These tasks are in charge of the communications with the outer layers of the architecture and with the computer decision models (Decision Modules). The Data Receiver task gathers the data received from the bracelet, using the SQLite database for storing the data. This task receives the block of the measurements sampled in the bracelet during one second as input; these blocks are securely stored in the SQLite database or data repository.

The Partitioning task is in charge of the offloading as well; its aim is to partition the data and request CC or MCC services with either raw data or processed data. Whenever there is unprocessed data, this task performs sliding windows on the data, requesting services for computing data transformations and for performing decision models based on data. According to the energy efficiency information, the task will request services to the MCC or to the CC, storing the intermediate data in the SQLite if needed. Besides, the Partitioning task should be divided into two parts: one is the windowing service and the other is just a job scheduling task, responsible for the CC/MCC services request as well. In parallel, an ontology of services should be developed and deployed into the SQLite database.

Furthermore, the software update task is responsible for deploying any update in both the models and their parameters and of the scheduling of new MCC services, new decision models or data processing; it is performed on demand of the CC layers. Coordination between this task and the standard app updating would eventually be needed.
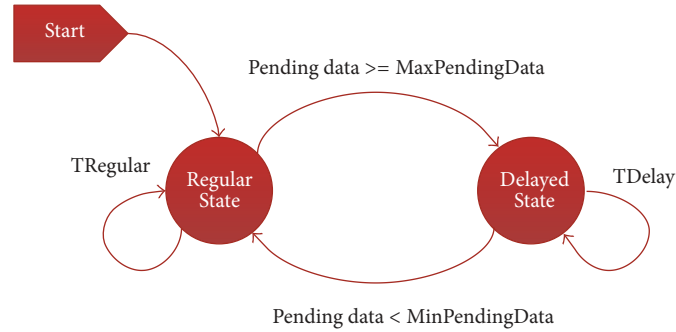
FIGURE 3: State diagram for the partitioning algorithm.

3.3.2. Reliability Issues in the Partitioning. This subsection focuses on three particular issues that IoT platforms must consider at least: the data delivery reliability, the data gathering service recovery, and the data delivery latency.

When talking about the two first issues, "reliability" refers to partitioning the data and requesting some services with the different partitions where the delivery of the data to the CC service must be guaranteed, while "recovery" deals with the healing of the crushed CC services and retrying the pending requests. Some of the solutions offered to cope with the reliability and the recovery of the services in the literature solve the problem at the expenses of the user experience [71], while other proposals like the Avatar system [73] suggest using a daemon that stays alive during network crashes keeping track of those partitions for which no service has been performed, retrying all the request just after the network connection is recovered. In this study, we propose an "Avatar"-like strategy, requesting the crashed services with the network recovery.

"Latency" is measured as the time elapsed since the request of a CC service on a partition until the answer or acknowledgment is received by the MCC. The performance of the system varies according to the data bunch size, ranging from 5 seconds to a few hours. Several factors have a great impact on the latency, for instance, the database's data storage mechanisms and the level of granularity of the services. Each of these factors needs further study in order to choose those that best fit the system performance in terms of real time and online monitoring. The experimentation in Section 4.2 analyzes the latency issue related to the MCC part.

MCC induces problems concerning the Quality of Service [17, 18]; the offloading techniques for partitioning are used to solve those problems related to the partitioning. Offloading varies from static partitioning of the data [74], aiming to reduce the battery consumption, to the dynamic partitioning [75, 76], where the partitioning is adjusted according to the network availability and the computing capacity of the mobile device. A hybrid partitioning is proposed using two defined data bunch sizes (NSRegular and NSDelayed). Each of them is used according to the network availability and its cost: if the WIFI networks are available then the NSRegular is used; otherwise the data is processed in the Smartphone and delivered in big bunches, of NSDelayed size, when the Wi-Fi network is present again. If the amount of undelivered data keeps growing up to MaxPendingData, then the data bunch size increases when Wi-Fi networks are available in order to deliver the data in the shortest period of time. Figure 3 shows the state diagram for the proposed partitioning.

The MCC computation and the data partitioning have a great impact on the energy expenditure and on the battery life: the smaller the computational effort in the Smartphone, the higher the network consumption. The reason for this compromise is that no computation in the MCC means delivering the instantaneous data to CC services; when using sampling frequencies higher than 1 Hz, the amount of data increases and so does the required communication bandwidth and complexity. Conversely, higher MCC computation induces a high reduction in the data that need CC delivery. Therefore, this research includes an energy efficiency experimentation to determine the best compromise in terms of how much computation can be offered by the MCC layer and how much are due to CC services.

The partitioning algorithm is described in Algorithm 1 where six global variables are set (lines (1)–(6)): DelayedState represents the state of working (initially it will be false), NSamples denotes the number of samples to send in each data bunch given the DelayedState, T is the delay time between two sent bunches, PendingData represents the number of samples to send, TempRepository is the local database, and finally WD represents the wearable device (Table 2 includes the description of the variables and values used in this algorithm). Two main tasks are scheduled (lines (7)-(8)): **DataReceiver** and **Partitioining** with a frequency of 1 and T seconds, respectively. The former deals with the communications with the bracelet, while the latter is devoted to the offloading and the data partitioning, including the partition type, RegularState or DelayedState. The DataReceiver running frequency is 1 second since it is the maximum frequency the WD can afford, while parameter T will be analyzed in Section 4.3.

The task **DataReceiver** (lines (10)–(19)) has to connect to WD through Bluetooth 4.0 with a discovering delay of TDiscovery seconds (line (16)). Once the connection is set, the data will be streamed from the WD with a frequency of one bunch per second. In parallel, the WD is sampling the HR and ACM sensors with a frequency of 16 Hz. In case the connection crashes, the lost samples will be skipped until the connection is recovered.

```
(1)    DelayedState ← false
(2)    NSamples ← NSRegular
(3)    T_DB ← TRegular
(4)    PendingData ← 0
(5)    WD ← Disconected
(6)    TempRepository ← []
(7)    Launch TimerTask DataReceiver() each 1 secs
(8)    Launch TimerTask Partitioining() each T_DB secs
(9)
(10)   function DATARECEIVER
(11)       if WD is connected then
(12)           Sample ← ReadDataFrom(WD)
(13)           PendingData ← PendingData + 1
(14)           TempRepository ← [TempRepository Sample]
(15)       else
(16)           Sleep(TDiscovery)
(17)           WD ← BlueToothDiscovery()
(18)       end if
(19)   end function
(20)
(21)   function PARTITIONING
(22)       if PendingData > MaxPendingData and delayed == false then
(23)           NSamples ← NSDelayed
(24)           T_DB ← TDelay
(25)           DelayedState ← true
(26)       else
(27)           if PendingData < MinPendingData and delayed == true then
(28)               NSamples ← NSRegular
(29)               T_DB ← TRegular
(30)               DelayedState ← false
(31)           end if
(32)       end if
(33)       if CurrentBatteryLevel < BatteryThreshold then
(34)           Bunch ← TempRepository[1 ⋯ NSamples]
(35)       else
(36)           Bunch ← CalcTransformations(TempRepository[1 ⋯ NSamples])
(37)       end if
(38)       Launch sent ← SendData(Bunch)
(39)       if sent == true then
(40)           PendingData ← PendingData − NSamples
(41)           TempRepository ← Remove(TempRepository, NSamples)
(42)       end if
(43)   end function
(44)
(45)   function SENDDATA(Bunch)
(46)       I ← NTry
(47)       sent ← false
(48)       while I < NTry and sent == false do
(49)           sent ← SendDataToCC(Bunch)
(50)           if sent == false then
(51)               Sleep(TRetry)
(52)           end if
(53)           return sent
(54)       end while
(55)   end function
```

ALGORITHM 1: Partitioning and offloading algorithm.

Table 2: Variables used in the partitioning algorithm.

Variable	Description
TDiscovery	The timeout in the process of finding the bracelet Bluetooth device when the connection gets lost.
NSamples	The number of samples to send in each data bunch depending on the state of working: (i) NSRegular: the number of samples to send in each regular data bunch. A regular data bunch size is defined with the samples to deliver, measured in seconds. For instance, if ACM and HR are sampled at 16 bits and 16 Hz, one sampled data includes the (a) 16 bits $*3$ axis = 6 bytes from the ACM, (b) the 16 bits (2 bytes) due to the HR, and (c) 16 bits (2 bytes) reserved for future extension and the timestamp in 8 bytes; then the samples in a second are $16 * 18 = 288$ bytes. If NSRegular is set to the samples for 60 seconds, the regular data bunch size is $60 * 288$ bytes. (ii) NSDelayed: the number of samples to send in delayed state should be bigger than the NSRegular in order to go back to the Regular state.
T_{DB}	The time between two data bunches depending on the state of working: (i) TRegular: the time between two regular bunches, that is, the time between launching SendData tasks for the Regular state. (ii) TDelay: the time between two delayed bunches, that is, the time between launching SendData tasks for the Delayed state.
PendingData	Number of samples pending to send to the CC services: (i) MaxPendingData: maximum number of data samples that have not been completely acknowledged by the CC service, making the state change from Regular to Delayed. (ii) MinPendingData: threshold to change from Delayed to Regular states.
NTry	The number of retries of the SendData task.
TRetry	Time between two consecutive SendData task retries.
DelayedHours	Time spent in Regular state, given in hours; afterwards the process gets into the Delayed state.

The task **Partitioining** (lines (21)–(43)) updates the bunch size (NSamples) depending on the PendingData. When this variable surpasses the MaxPendingData threshold, the DelayedState is set to true and the related variables are updated. When PendingData is below MinPendingData, the DelayedState is set to false. Next, if CurrentBatteryLevel exceeds BatteryThreshold then the transformations of the first pending bunch in the TempRespository are calculated; see Section 4.1; otherwise, it is the raw data that is sent instead of its transformations (lines (33)–(37)). The bunch is sent to the CC services using the **SendData** function (line (38)) and in this case only the process succeeds, the PendingData is updated, and the bunch is removed from the TempRepository (lines (39)–(42)).

The auxiliary function **SendData** (lines (45)–(55)) will try to send the bunch to the CC services NTry times with a timeout of TRetry seconds (line (51)) between runs.

3.3.3. Security Issues within the Platform. Several ad hoc frameworks have been reported in the literature concerning interfaces between mobile devices and cloud services. To our knowledge, these solutions are not suitable for the specific problem focused on this research. Some of these frameworks, through BNS and IoT ad hoc solutions, are not secure [71, 77]; furthermore, its focus is a general platform, making it difficult to be extended for specific solutions [11]. Other solutions, like those in [70, 78], though being secure ad hoc frameworks, make use of weblets; therefore, the increased computational cost makes them fruitless [12]. Concerning the epilepsy specific frameworks, those reported in the literature are either very specific or closed solutions [7, 41], being isolated and storing all the patients' data in the Smartphone [32]. Thus,

it was decided to develop our own framework to fit the specific needs of the problem, considering the extensibility and enhancement issues to improve the development.

The proposed architecture involves two kinds of vulnerable wireless connections: a bluetooth connection between the WD and the mobile phone and a wireless connection (Wi-Fi or data) between the mobile phone and the CC based on the REST protocol. The security and privacy issues on wearable communications are a challenging field of study [79, 80] since the computing capability and battery life of these devices are quite limited. In this sense, one of the most popular techniques for key generation/key agreement for wearable devices is based on physiological print such as the interpulse interval (IPI) [81]; however, our first version of WD does not include any kind of extra security issues since it will used in a secure and controlled environment for the first clinical tests. Besides, the wireless connection between the mobile phone and the CC is carried out using a RESTFul service together with the HTTP communication protocol [82], since we know the optimal solution would be an HTTPs connection.

3.3.4. Detection of Epileptic Seizures. Two different types of models are proposed for the detection of focal myoclonic epileptic seizures: on the one hand, Genetic Fuzzy Finite State Machines (GFFSM) applied to the epilepsy recognition [3] and, on the other hand, a feature extraction using a Distance-based Principal Component Analysis (DPCA) step followed by a K-Nearest Neighbor (KNN) classifier [65].

GFFSM defines two main fuzzy sets to describe the current state (seizure or normal) and a set of four fuzzy rules (IF-THEN fuzzy rules) whose output is the new fuzzy state. The ACM values are transformed to three new variables, Signal

TABLE 3: Bracelet and Smartphone specifications.

	Bracelet		Smartphone
Microcontroller	SoC BTLE nRF51822 (32-bit ARM Cortex M0 core) 256 kB RAM 32 kB FLASH	Platform and OS	32 bits, MediaTek MT6582, Android Lolipop 5.1
Accelerometry sensor	3-axial, 2G AD ADXL327BCPZ	CPU and core	1.3 Ghz Quad-Core ARM Cortex-A7
Pulse monitor	TI AFE4400RHAT	GPU	ARM Mali-400 MP2 500 Mhz
Pulse sensor	LED verde OSRAM SFH7050	RAM	1 GB
Battery	Ion-Lithium 425 mAh	Battery	Li-Po 2150 mAh
Antenna	PCB 2.4 GHz	Storage	8 GB

Magnitude Ratio (SMA), Amount of Movement (AoM), and Time between Peaks (TbP), which are the input variables, together with the current fuzzy State, to the fuzzy rule system. Provided that a good variable fuzzy partitioning algorithm is used [83], the GFFSM method produces highly generalized models to cope with a wide population [3].

Besides, the method published in [65] makes use of the ACM values, computing up to 23 different transformations, SMA, AoM, and TbP, among others, but transforms the domain to another one using DPCA. DPCA hybridizes Locally Linear Embedding (LLE) with Principal Component Analysis: the distance matrix is used to perform the PCA transformation instead of the covariance. In addition, the number of desired features in the transformed domain is given a priori, like in LLE; therefore high reduction in the dimensionality can be obtained. Applying DPCA to the dataset of all the known transformations for the ACM values, 23 features, and afterwards a K-NN classifier, with K set to 3, led to results similar to those obtained for GFFSM explained above.

These two options represent two totally different approaches: the former, with a reduced set of rules and states, stands for general models, valid for a wide population, that introduces simple computations; thus they can be easily performed in the MCC size. The main drawback of this method is the learning stage, which cannot be performed in MCC; however, some tuning and active learning issues can be considered in this context. On the other hand, DPCA + KNN introduces much more computation requirements in the detection service, but the learning and updating are absolutely affordable in terms of computational cost.

4. Numerical Results

This section deals with the studies concerning the latency and the energy efficiency described in the previous section (see Section 3). The aim of this experimentation is twofold: on the one hand, to determine the data bunch size for each of the states by means of analyzing the induced latency; on the other hand, the impact of MCC versus CC offloading in the duty cycle of the battery in a real context, so to choose the best energy efficiency balance.

The organization of this section is as follows. Firstly, the materials and methods used in this experimentation are detailed. Afterwards, the latency study is included in Section 4.2. Finally, the energy efficiency issues are presented in Section 4.3. In each of these subsections, a discussion on the findings is included.

4.1. Material and Methods. To carry on the experimentation, part of the detailed PK has been developed using an ad hoc bracelet; the core of the CC layer has also been implemented. The whole system is able to monitor the patient's behaviour, storing the data and performing transformations of the measured and sampled physical variables.

The ad hoc bracelet has been developed at the Instituto Tecnológico de Castilla y León [84], though the whole team participated in the design. It includes an Analog Devices ACM sensor and a Texas Instruments green light LED HR sensor; however, with the development of the wearable techniques, new sensors would eventually be introduced as well, extending the frontiers of the seizures and events to discover.

This bracelet delivers the sampled data to the linked Smartphone using bluetooth 4.0 Low Energy; the sampling frequency for the ACM has been set to 16 Hz, while the HR is based on reflected light and the measurements are gathered on demand with at least 10 seconds between consecutive estimations. The WD includes a 420 mA battery, which beats plenty of the solutions in the market and allows a day-long duty cycle The MCC is implemented using a centralized approach [11], which is depicted in Figure 4.

The Smartphone is a mid-class Android mobile device, with bluetooth 4.0 Low Energy capabilities. Obviously, the mobile is also provided with a Wi-Fi connection. The experimentation has been performed in a laboratory including two PKs, a Wi-Fi router, and a low-cost server located in a different laboratory, actually, in a another city, the two laboratories connected to the Internet. Table 3 includes the data specifications of the WD and the Smartphone, while Table 4 shows the specifications of the deployed server.

The MCC layer incorporates the Data Receiver and the Partitioning tasks, makes use of the SQLite database, and can accept requests to compute the ACM and HR defined transformations. Whenever the battery level is under a predefined
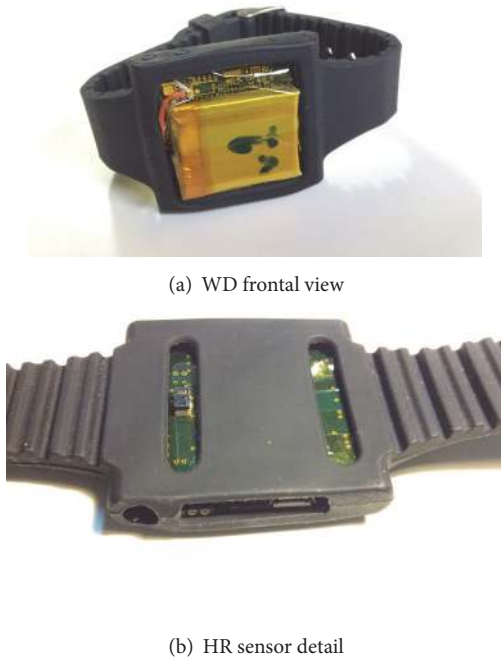
(a) WD frontal view

(b) HR sensor detail

FIGURE 4: (a) The ad hoc developed WD is shown, while (b) the green light LED on the bottom of the bracelet is shown.

```
(1)  .....
(2)  while I < NTry and sent == false do
(3)      Time_tic_start ← GetMilliseconds()
(4)      sent ← SendDataToCC(Bunch)
(5)      Time_tic_end ← GetMilliseconds()
(6)      Latency ← Time_tic_end – Time_tic_start
(7)      if sent == false then
(8)          Sleep(TRetry)
(9)      end ifreturn sent
(10) end while
(11) .....
```

ALGORITHM 2: Excerpt from partitioning main algorithm with the latency calculation specification.

TABLE 4: CC Server side specifications.

Server	HP ProLiant DL320 G4 2 GB RAM (Dual-Core Pentium D 900)
OS	Scientific Linux 6.3
HTTP server	Apache 2
REST services framework	SlimPHP 0.9
Database storage	QNAP TS412U, 4 × 1 TB HD, RAID1
Database server	MySQL 5.1.73, database on RAID1 with granted access through NFS

threshold, the raw data will be stored in the local database, requesting the further services when this condition is not held anymore. The CC layer can accept requests for storing and for processing data. Besides, a data unit sent from the bracelet to the Smartphone includes three axial components from ACM, the HR, an extra field, and the time stamp. All of them except the time stamp are 16 bits long; the time stamp is stored in 32 bits. The predefined transformations used in this experimentation include the SMA, the AoM, and the TbP, computed as reported in [3].

Two possible scenarios will be considered: the first, sending the raw data from the MCC to the CC; the second, performing the preprocessing in the MCC and sending the transformations to the CC. In both cases, JSON messages will be interchanged. Whenever raw data is interchanged between the MCC and the CC layers the JSON message is structured as shown in the upper part of Table 5, using the integer representation of the sampled physical variables.

A data bunch including samples for periods of 30 seconds long, therefore, will be of the size of 51,360 bytes.

Whenever the data sent from the MCC layer to the CC layer includes the transformations, the JSON format is as shown in the bottom part of Table 5. In this case, if a window size of 2 seconds is used, the data bunch would include 2755 bytes.

Finally, two experiments have been carried out. The former is related to analyzing the latency in the two defined states, RegularState and DelayedState, for different data bunch sizes when sending raw data from the MCC to the CC, with the aim of finding the best balance. The latency is measured during the data transmission as stated in Algorithm 2, which is an excerpt of the algorithm shown in partitioning but extended to include the measurement of the latency. Different sizes have been tested: 6.80 KB (samples for 5 seconds, a total of 80 raw samples), 13.59 KB (10 s, 160 raw samples), 20.39 KB (15 s, 240 raw samples), 40.78 KB (30 s, 480 raw samples), 81.56 K (1 min, 960 raw samples), 407.81 KB (5 min, 4800 raw samples), 2446.88 KB (30 min, 28800 raw samples), and 4893.75 KB (60 min, 57600 raw samples). Ten repetitions of the test have been run to obtain the statistical performance.

Once the best data bunch size is found, the second experiment aims to evaluate the best performance between MCC preprocessing and CC preprocessing. In the former case, the MCC layer computes the transformations and sends them in data bunches, while in the latter raw data is sent. In both cases, the same data bunch size is used: the one found optimum in the first experiment. The PK performs its normal operation from full charge to total discharge. Two series of runs have been run: one sending raw data to the CC and the other computing the transformations at MCC level and sending these transformations. Again, ten runs for each series have been performed. The next two subsections show the discussion on the obtained results for each experiment.

4.2. Discussion on the Results for the Latency Test. Results are shown in Table 6 and in Figures 5 and 6. Table 6 shows the obtained times of latency in milliseconds for each of the 10 runs of the experiment and for each data bunch size. Moreover, the mean and the standard deviation are also

TABLE 5: JSON interchanged messages. The upper message has been used when sending raw data, while the bottom one has been used when sending the transformations.

```
[
{"IDPatient":"1","Time":1484912812,"HR":"0","RESERVE":"0","ACCX":"399","ACCY":"413","ACCZ":"456"},
{"IDPatient":"1","Time":1484912812,"HR":"0","RESERVE":"0","ACCX":"399","ACCY":"413","ACCZ":"456"},
{"IDPatient":"1","Time":1484912812,"HR":"0","RESERVE":"0","ACCX":"399","ACCY":"413","ACCZ":"456"},
..... ]
[
{"IDPatient":"1","Time":1484912812,"HR":"98","Oxigen":"0","SMA":"1.98","AoM":"3.23","TbP":"4.56"}
]
```

TABLE 6: Latency obtained results, measured in milliseconds, for each data bunch size and iteration. The four bottom rows are the main statistics, the mean and the standard deviation, the time between data bunches (T_{DB}), and the ratio latency time versus T_{DB}.

| Number of samples	80	160	240	480	960	4800	28800	57600
KB	6.80	13.59	20.39	40.78	81.56	407.81	2446.88	4893.75
Iter								
1	191	318	456	875	2082	8994	53234	105975
2	185	325	452	1116	2043	9973	52927	104775
3	155	305	467	1168	2057	9055	52594	110677
4	180	327	468	900	1967	9145	52296	105161
5	186	300	458	890	1844	9537	52585	106078
6	152	305	474	907	1965	9538	52619	105476
7	177	337	504	1154	1843	8916	52206	105004
8	182	565	464	1145	1956	8874	52413	104416
9	165	322	600	885	1825	8874	53141	104864
10	177	325	458	907	2083	8862	52312	104522
Mean	175,00	342,9	480,10	994,70	1966,50	9176,80	52632,7	105694,80
Std	13,28	78,90	44,61	130,97	100,75	379,17	340,15	1838,02
Latency/T_{DB}	0.035	0.034	0.032	0.033	0.033	0.031	0.029	0.029
T_{DB}	5	10	15	30	60	300	1800	3600
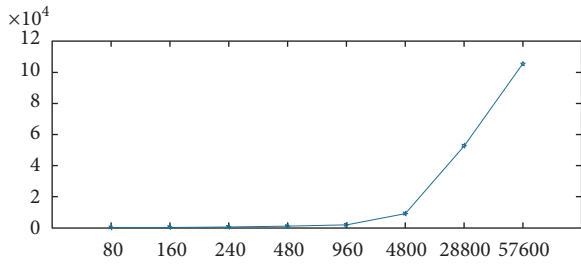
FIGURE 5: The exponential relationship between the data bunch size, x-axis, in KB, and the mean latency time, y-axis, in milliseconds, for the CC processing mode.
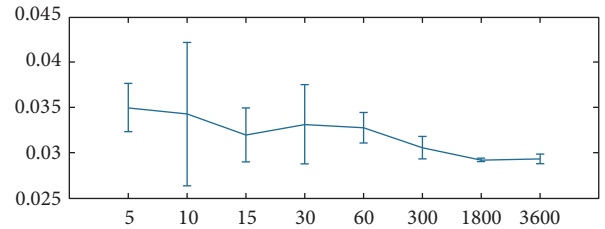

FIGURE 6: The relationship between the data bunch size and the ratio latency time versus the time between data bunch generation for the CC preprocessing mode. Some smaller sizes exhibit a wide spread; the higher the ratio, the narrower the variation in the performance.

calculated and shown for each size over the 10 runs. Finally, the time between consecutive data bunches (T_{DB}) and the ratio of the latency time versus T_{DB} are shown.

The relationship between the mean of the latency time and the data bunch size is shown in Figure 5, where it clearly shows two sizes as candidates: 20.39 KB, equivalent to 15 s gathering 240 raw samples, and 40.78 KB, equivalent to 30 s gathering 480 raw samples, represent the best compromise between the communication acts and the response time.

Smaller sizes might introduce a shorter response but the communication acts will have a high impact on the battery life. Furthermore, the mean latency times, 480.10 ms and 994.70 ms, correspondingly, are still negligible compared to the time between consecutive bunches, 15 s and 30 s, allowing the recovery of connection lost events easily. More specifically, up to 14 retries can be securely done with a safe waiting period of 0.5 s (1 s for the second option) without overlapping, which can be considered enough in a real context.

TABLE 7: Performance facts when comparing MCC with CC services for the two analyzed data bunch sizes, 20.39 KB and 40.78 KB. For the 20.39 KB size, only the MCC has been performed as detailed in the text.

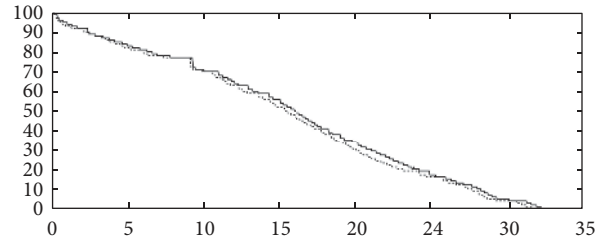| Parameter	20.39 KB	40.78 KB	
	MCC	MCC	CC
Battery life time (h)	28.998	32.232	31.600
MB of transferred data	6.625	8.702	159.223
Latency time (h)	4.491	4.555	51.214
Computational time (h)	1.041	1.395	0.000
Latency + computation (h)	5.532	5.950	51.214

FIGURE 7: Comparison of the Smartphone's battery discharge when using the data bunch size of 20.39 KB for the two options: MCC preprocessing versus CC preprocessing. The continuous line stands for the MCC preprocessing mode, while the dotted line stands for the CC preprocessing mode. On the y-axis, the battery charge level is in percentage. The x-axis shows the time in hours.
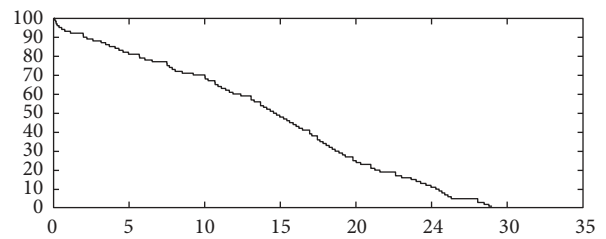

FIGURE 8: Comparison of the Smartphone's battery discharges when using the data bunch size of 40.78 KB for the MCC preprocessing. On the y-axis, the battery charge level is in percentage. The x-axis shows the time in hours.

These findings are also remarked in Figure 6. This figure shows the ratio latency time versus T_{DB} for all the iteration on each different data bunch size. For the smaller sizes, a high ratio variability is observed, which means that the reliability of the data delivery is in compromise. Higher values induce reduced risks, but at a cost of real time response. Considering the previous results and the ratio in this latter table, the data bunch size of 20.39 KB represents the best compromise.

4.3. Battery Duty Cycle Test. This test proposes putting the PK into deployment in the two main cases: with MCC computation and with all the computations carried out in the CC layer. In the former case, the communication acts deliver the transformation of the raw data, reducing the amount of data to send; the latter case includes delivering the raw data, reducing the computation at the expenses of increasing the amount of data to deliver. Two data bunch sizes are analyzed and used in this experimentation, those that found the best solutions in the previous subsection: 20.39 KB and 40.78 KB. We performed the CC and MCC calculations for 40.78 KB first; according to the obtained results, only the MCC calculations have been considered for the 20.39 KB case.

The results for these performance tests are included in Table 7 and Figures 7 and 8. Clearly, the main contribution to the battery consumption is the communication acts: the higher the computation in the Smartphone, the better. This conclusion only holds for those components that do not include human-machine interface, like the touchscreen. Besides, further analysis is required in order to evaluate the complexity of the calculations that do not introduce extra battery consumption. Finally, according to the evolution of the battery duty cycle, the most interesting data bunch size is 40.78 KB, as long as it provides the longest battery discharge time, while performing similarly concerning the latency. However, the 20.39 KB size can be considered valid as well, as no main differences have been found.

4.4. Deployment Cost of the Platform. The whole approach is based on relatively low-cost elements: a mid-range Smartphone plus a WD, the PK, worth less than 500 euros to the

health system and even less if we consider that the majority of the population already owns a valid Smartphone. The deployment of the CC services can be done either externally or in their own data center; the balance between MCC and CC services might help in reducing this deployment cost. Besides, the cost of Smartphones with the minimum requirements is normally 130 euros at the most. On the other hand, the wristband used in this study is worth 300 euros, but this price can be highly reduced; commercial Smartwatches integrating the HR and ACM are available from 150 euros.

It is worth mentioning that one of the main concerns with the wristband selection is the battery life: this parameter must be higher than 24 hours. A smaller battery life could lead to problems like a high number of charging cycles or too short a working time between charges. In any case, there is always the possibility of the battery running out, but this risk gets higher with reduced battery life periods. And this is something that can make the whole platform useless, increasing its opportunity cost.

Moreover, the integration of federated CC servers in outpatient clinics, even in family homes, with local servers under 500 euros nowadays, will highly decrease the amount of power required by the health system and outperforming both the robustness and the real time response. Not related to IoT but to eHealth, such a distributed solution has been successfully used in GNU Health [85]. This cost analysis has been barely performed; and therefore, the main part of the solutions is either expensive or uncomfortable, or even

both at the same time. Each design decision has been taken considering the cost analysis of the candidate solutions.

5. Conclusions

This study analyzes the solutions in the literature describing solutions for epilepsy tonic-clonic seizure detection and monitoring. The majority of the approaches lack several remarkable factors: developing ergonomic approaches, supporting everyday life, providing economical affordable solutions, introducing storage of the sampled data and providing intelligent CC services, introducing real time response, or considering multiple services on the MCC size. This study addresses the design of an IoT platform for the epilepsy seizure detection and monitoring considering each of these factors.

The solution is based on a WD to be located on a wrist connected to a Smartphone, which in turns implements MCC services and has access to CC services as well. The global goal is detecting the seizures, storing information from the sensory system, generating alarms and notifications, performing machine learning techniques on the data to learn the best models to detect or to visualize the data, sharing data, and providing processed information to the medical staff, among others. Special attention has been paid to the MCC module, where some design decisions are discussed, leading to the experimentation stage.

The experimentation stage implemented part of the MCC and CC modules, developing an ad hoc solution for the WD. The experimentation has been focused on determining the best data bunch size and on drawing conclusions concerning the criteria to choose when performing computation on the MCC versus requesting services on raw data to the CC layer. The experimentation results show two possible data bunch sizes (20.39 and 40.78 KB) as the most suitable ones. Furthermore, the second stage of the experimentation suggests that plenty of computation can be delivered on the Smartphones, reducing the amount of networking. Furthermore, special care should be taken to reduce the power consumption due to some mobile components, such as touchscreens. This research is only in its early stages, and in the near future we expect to complete the design, considering the integration of this framework into publicly available open software health platforms, such as GNU Health.

Abbreviations

ACM: Triaxial accelerometer
AoM: Amount Of Movement
BSN: Body Sensor Networks
CC: Cloud Computing
DPCA: Distance-based Principal Component Analysis
EEG: Electroencephalogram
ECG: Electrocardiogram
GFFSM: Genetic Fuzzy Finite State Machine
HR: Heart rate
KNN: K-Nearest Neighbor
MANET: Mobile Ad Hoc NETworks
MCC: Mobile Cloud Computing
PK: Patient's kit
SMA: Signal Magnitude Ratio
TbP: Time between Peaks
T_{DB}: Time between consecutive data bunches
WD: Wearable device.

Conflicts of Interest

The authors declare that they have no conflicts of interest.
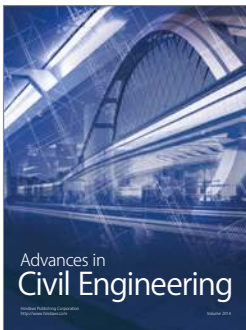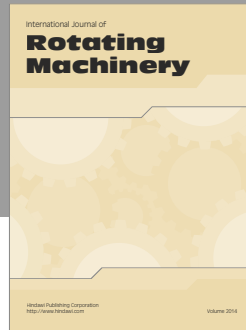
Acknowledgments

References

[1] R. S. Fisher, W. Van Emde Boas, W. Blume et al., "Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)," *Epilepsia*, vol. 46, no. 4, pp. 470–472, 2005.

[2] J. Engel Jr., "A proposed diagnostic scheme for people with epileptic seizures and with epilepsy: report of the ILAE task force on classification and terminology," *Epilepsia*, vol. 42, no. 6, pp. 796–803, 2001.

[3] J. R. Villar, P. Vergara, M. Menéndez, E. De La Cal, V. M. González, and J. Sedano, "Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition," *International Journal of Neural Systems*, vol. 26, no. 6, 2016.

[4] E. B. Petersen, J. Duun-Henriksen, A. Mazzaretto, T. W. Kjar, C. E. Thomsen, and H. B. D. Sorensen, "Generic single-channel detection of absence seizures," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '11)*, pp. 4820–4823, Boston, Mass, USA, September 2011.

[5] C. Pradhan, S. Sinha, K. Thennarasu, and T. Jagadisha, "Quantitative analysis of heart rate variability in patients with absence epilepsy," *Neurology India*, vol. 59, no. 1, pp. 25–29, 2011.

[6] S. Beniczky, T. Polster, T. W. Kjaer, and H. Hjalgrim, "Detection of generalized tonic-clonic seizures by a wireless wrist accelerometer: a prospective, multicenter study," *Epilepsia*, vol. 54, no. 4, pp. e58–e61, 2013.

[7] D. Callegari, E. Conte, T. Ferreto et al., "EpiCare - a home care platform based on mobile cloud computing to assist epilepsy diagnosis," in *Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare, MOBIHEALTH 2014*, pp. 148–151, November 2014.

[8] P. Lokhande and T. Mote, "Epilepsy monitoring and analysis system using android platform," *International Journal of Sciences and Research*, vol. 5, no. 7, pp. 60–64, 2016.

[9] P. Meghana and V. V. Sravani, "Mobile application based detection of seizures using inertial measurement units and emergency care," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 11, pp. 6388–6391, 2015.

[10] S. Ramgopal, S. Thome-Souza, M. Jackson et al., "Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy," *Epilepsy and Behavior*, vol. 37, pp. 291–307, 2014.

[11] S. Deshmukh and R. Shah, "Computation offloading frameworks in mobile cloud computing: a survey," in *Proceedings of the 2016 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2016*, pp. 1–5, March 2016.

[12] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[13] M. Othman, S. A. Madani, S. U. Khan et al., "A survey of mobile cloud computing application models," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 393–413, 2014.

[14] M. M. Hassan, H. S. Albakr, and H. Al-Dossari, "A cloud-assisted Internet of Things framework for pervasive healthcare in smart city environment," in *Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities, EMASC 2014*, pp. 9–13, ACM.

[15] G. Fortino, M. Pathan, and G. di Fatta, "BodyCloud: integration of cloud computing and body sensor networks," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 851–856, IEEE, December 2012.

[16] A. Khelil, F. K. Shaikh, A. A. Sheikh, E. Felemban, and H. Bojan, "DigiAID: A wearable health platform for automated self-tagging in emergency cases," in *Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare, MOBIHEALTH 2014*, pp. 296–299, IEEE, November 2014.

[17] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[18] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: a survey, state of art and future directions," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 133–143, 2014.

[19] C. Doukas and I. Maglogiannis, "Managing wearable sensor data through Cloud Computing," in *Proceedings of the 2011 3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2011*, pp. 440–445, IEEE, December 2011.

[20] C. Doukas and I. Maglogiannis, "Bringing IoT and cloud computing towards pervasive healthcare," in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012*, pp. 922–926, IEEE, July 2012.

[21] A. Forkan, I. Khalil, and Z. Tari, "CoCaMAAL: a cloud-oriented context-aware middleware in ambient assisted living," *Future Generation Computer Systems*, vol. 35, pp. 114–127, 2014.

[22] M. Chen, Y. Ma, S. Ullah, W. Cai, and E. Song, "ROCHAS: Robotics and cloud-assisted healthcare system for empty nester," in *Proceedings of the 8th International Conference on Body Area Networks, BODYNETS 2013*, pp. 217–220, October 2013.

[23] C. Doukas, T. Pliakas, and I. Maglogiannis, "Mobile healthcare information management utilizing cloud computing and Android OS," in *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '10)*, pp. 1037–1040, IEEE, September 2010.

[24] X. Huang and P. M. Matricardi, "Allergy and asthma care in the mobile phone era," *Clinical Reviews in Allergy and Immunology*, pp. 1–13, 2016.

[25] X. Liu, R. Wang, D. Zhou, and Z. Hong, "Smartphone applications for seizure care and management in children and adolescents with epilepsy: Feasibility and acceptability assessment among caregivers in China," *Epilepsy Research*, vol. 127, pp. 1–5, 2016.

[26] M. Chen, Y. Zhang, Y. Li, M. M. Hassan, and A. Alamri, "AIWAC: affective interaction through wearable computing and cloud technology," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 20–27, 2015.

[27] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[28] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews*, vol. 40, no. 1, pp. 1–12, 2010.

[29] B. Direito, C. Teixeira, B. Ribeiro, M. Castelo-Branco, F. Sales, and A. Dourado, "Modeling epileptic brain states using EEG spectral analysis and topographic mapping," *Journal of Neuroscience Methods*, vol. 210, no. 2, pp. 220–229, 2012.

[30] S. Xie and S. Krishnan, "Wavelet-based sparse functional linear model with applications to EEGs seizure detection and epilepsy diagnosis," *Medical and Biological Engineering and Computing*, vol. 51, no. 1-2, pp. 49–60, 2013.

[31] S. Sareen, S. K. Sood, and S. K. Gupta, "An automatic prediction of epileptic seizures using cloud computing and wireless sensor networks," *Journal of Medical Systems*, vol. 40, no. 11, article 226, 2016.

[32] S. Sareen, S. K. Sood, and S. K. Gupta, "A cloud-based seizure alert system for epileptic patients that uses higher-order statistics," *Computing in Science and Engineering*, vol. 18, no. 5, pp. 56–67, 2016.

[33] A. Ulate-Campos, F. Coughlin, M. Gaínza-Lein, I. S. Fernández, P. L. Pearl, and T. Loddenkemper, "Automated seizure detection systems and their effectiveness for each type of seizure," *Seizure*, vol. 40, pp. 88–101, 2016.

[34] N. A. Marzuki, W. Husain, and A. M. Shahiri, "Myepipal, Mobile application for managing, monitoring and predicting epilepsy patient," in *Proceedings of the International Conference on Advances in Information and Communication Technology*, pp. 383–392, Springer, 2016.

[35] P. G. K. Prince, R. Hemamalini, and R. Immanuel Rajkumar, "LabVIEW based abnormal muscular movement and fall detection using MEMS accelerometer during the occurrence of seizure," *Indian Journal of Science and Technology*, vol. 7, no. 10, pp. 1625–1631, 2014.

[36] E. Schulc, I. Unterberger, S. Saboor et al., "Measurement and quantification of generalized tonic-clonic seizures in epilepsy patients by means of accelerometry-An explorative study," *Epilepsy Research*, vol. 95, no. 1-2, pp. 173–183, 2011.

[37] T. E. society, *Free Epilepsy Smart-Phone App*, 2013, https://www.epilepsysociety.org.uk/free-epilepsy-smartphone-app#.WC1jamSLR8c.

[38] V.-M. Teohari, C. Ungureanu, V. Bui, J. Arends, and R. M. Aarts, "Epilepsy seizure detection app for wearable technologies," in *Proceedings of the European Conference on Ambient Intelligence*, 2014.

[39] F. Masse, M. V. Bussel, A. Serteyn, J. Arends, and J. Penders, "Miniaturized wireless ECG monitor for real-time detection of epileptic seizures," *Transactions on Embedded Computing Systems*, vol. 12, no. 4, article 102, 2013.

[40] M. D. Kamat, M. P. S. Ganorkar, and M. R. Jain, "Child activity monitoring using sensors," *International Journal of Engineering and Techniques*, vol. 1, no. 3, pp. 129–133, 2015.

[41] G. Bajwa, R. Dantu, M. F. M. Issadeen, and R. M. Joseph, "Self-tracking via brain-mobile-cloud interface," in *Proceedings of the AAAI Spring Symposium Series*, 2013.

[42] K. G. Raam and R. Sasikala, "Detection and prediction of seizures using a wrist-based wearable platform," *International Journal of Control Theory and Applications*, vol. 9, no. 37, pp. 193–203, 2016.

[43] R. Sasikala, A. Asuntha, and S. Indirani, "Detection and prediction of seizures using a wrist-based wearable platform," *Journal of Chemical and Pharmaceutical Sciences*, vol. 9, no. 4, pp. 3208–3215, 2016.

[44] M. Velez, R. S. Fisher, V. Bartlett, and S. Le, "Tracking generalized tonic-clonic seizures with a wrist accelerometer linked to an online database," *Seizure*, vol. 39, pp. 13–18, 2016.

[45] H. Goelz, R. D. Jones, and P. J. Bones, "Wavelet analysis of transient biomedical signals and its application to detection of epileptiform activity in the EEG," *Clinical EEG and Neuroscience*, vol. 31, no. 4, pp. 181–191, 2000.

[46] M. Milošević, A. Van De Vel, B. Bonroy et al., "Detection of epileptic convulsions from accelerometry signals through machine learning approach," in *Proceedings of the 2014 24th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2014*, pp. 1–6, IEEE, September 2014.

[47] A. Van de Vel, K. Cuppens, B. Bonroy et al., "Long-term home monitoring of hypermotor seizures by patient-worn accelerometers," *Epilepsy and Behavior*, vol. 26, no. 1, pp. 118–125, 2013.

[48] A. Van de Vel, M. Milosevic, B. Bonroy et al., "Long-term accelerometry-triggered video monitoring and detection of tonic-clonic and clonic seizures in a home environment: pilot study," *Epilepsy and Behavior Case Reports*, vol. 5, pp. 66–71, 2016.

[49] Y. S. Ding and Z. Zilic, "ECG compression for mobile sensor platforms," in *Proceedings of the 13th Annual Body Sensor Networks Conference, BSN 2016*, pp. 99–104, June 2016.

[50] Neurosky. Eeg, neurosky mindband. 2009 http://store.neurosky.com/.

[51] Epdetect. A free mobile application to detect epileptic seizures with automated carer alert, 2009 http://www.epdetect.com/.

[52] R. S. Fisher, E. Bartfeld, and J. A. Cramer, "Use of an online epilepsy diary to characterize repetitive seizures," *Epilepsy and Behavior*, vol. 47, pp. 66–71, 2015.

[53] L. Irody, *Mobile Patient Diaries: Epidiary*, 2007, http://www.irody.com/mobile-patient-diaries/.

[54] Jiwok, *Epi & Me*, 2014, http://www.epiandme.com/.

[55] L. N. Ranganathan, S. Aadhimoolam Chinnadurai, B. Samivel, B. Kesavamurthy, and M. M. Mehndiratta, "Application of mobile phones in epilepsy care," *International Journal of Epilepsy*, vol. 2, no. 1, pp. 28–37, 2015.

[56] U. K. D. Roman and H. Lausen, *The Web Service Modeling Ontology Wsmo, Final Version*, 2005, Online; accessed 3-May-2017 http://www.wsmo.org/TR/d2/v1.1/.

[57] M. S. John Domingue and D. Roman, Web service modeling ontology (wsmo) - an ontology for semantic web services. 2005. Online; accessed 3-May-2017.

[58] D. Roman, U. Keller, H. Lausen et al., "Web service modeling ontology," *Applied ontology*, vol. 1, no. 1, pp. 77–106, 2005.

[59] M. Aazam, E. Huh, M. St-Hilaire, C. Lung, and I. Lambadaris, "Cloud of Things: Integration of IoT with Cloud Computing," in *Robots and Sensor Clouds*, vol. 36 of *Studies in Systems, Decision and Control*, pp. 77–94, Springer International Publishing, Cham, Switzerland, 2016.

[60] T. C. Arcadius, B. Gao, G. Y. Tian, and Y. Yan, "Structural health monitoring framework based on internet of things: a survey," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 619–635, 2017.

[61] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of things: a review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, article 1069, 2016.

[62] M. Hassanalieragh, A. Page, T. Soyata et al., "Health monitoring and management using internet-of-things (IoT) sensing with cloud-based processing: opportunities and challenges," in *Proceedings of the IEEE International Conference on Services Computing, SCC 2015*, pp. 285–292, IEEE, July 2015.

[63] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[64] IBM. What is mobile cloud computing? 2017. Online; accessed 5-May-2017 https://www.ibm.com/cloud-computing/learn-more/what-is-mobile-cloud-com%20puting/.

[65] J. R. Villar, M. Menéndez, E. de la Cal, J. Sedano, and V. M. González, "Identification of abnormal movements with 3D accelerometer sensors for seizure recognition," *Journal of Applied Logic*, 2016.

[66] J. Alcalá-Fdez, A. Fernández, J. Luengo et al., "KEEL datamining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.

[67] I. H. W. Eibe Frank and M. A. Hall, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, Morgan Kaufmann, 4th edition, 2016.

[68] F. Liu, P. Shu, H. Jin et al., "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–21, 2013.

[69] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[70] D. Huang, X. Zhang, M. Kang, and J. Luo, "MobiCloud: building secure cloud framework for mobile computing and communication," in *Proceedings of the 5th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2010*, pp. 27–34, IEEE, June 2010.

[71] R. Hasan, M. M. Hossain, and R. Khan, "Aura: An IoT based cloud infrastructure for localized mobile computation outsourcing," in *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2015*, pp. 183–188, April 2015.

[72] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond (MCS '10)*, ACM, June 2010.

[73] C. Borcea, X. Ding, N. Gehani, R. Curtmola, M. A. Khan, and H. Debnath, "Avatar: mobile distributed computing in the cloud," in *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2015*, pp. 151–156, April 2015.

[74] E. Cuervoy, A. Balasubramanian, D.-K. Cho et al., "MAUI: making smartphones last longer with code offload," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 49–62, New York, NY, USA, June 2010.

[75] P. Angin, B. Bhargava, and Z. Jin, "A self-cloning agents based model for high-performance mobile-cloud computing," in *Proceedings of the 8th IEEE International Conference on Cloud Computing, CLOUD 2015*, pp. 301–308, July 2015.

[76] H. Wu and D. Huang, "MoSeC: Mobile-cloud service composition," in *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Mobile-Cloud 2015*, pp. 177–182, IEEE, April 2015.

[77] G. Fortino, D. Parisi, V. Pirrone, and G. Di Fatta, "BodyCloud: a SaaS approach for community body sensor networks," *Future Generation Computer Systems*, vol. 35, pp. 62–79, 2014.

[78] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using mapreduce," Technical Report, DTIC Document, 2009.

[79] X. Bellekens, A. Hamilton, P. Seeam, K. Nieradzinska, Q. Franssen, and A. Seeam, "Pervasive eHealth services a security and privacy risk awareness survey," in *Proceedings of the 2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2016*, June 2016.

[80] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *Proceedings of the 52nd Annual Design Automation Conference, DAC '15*, pp. 54:1–54:6, ACM, New York, NY, USA, 2015.

[81] S. Wang, R. Bie, F. Zhao, N. Zhang, X. Cheng, and H.-A. Choi, "Security in wearable communications," *IEEE Network*, vol. 30, no. 5, pp. 61–67, 2016.

[82] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, pp. 627–633, ACM, October 2009.

[83] J. R. Villar, P. Vergara, E. D. La Cal, M. Menéndez, and J. Sedano, "Pre-clinical study on the detection of simulated epileptic seizures," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 24, no. Suppl. 2, pp. 33–46, 2016.

[84] Applied electronics and artificial intelligence department; Instituto Tecnológico de Castilla y León, 2017, http://www.itcl.es/.

[85] GNU health - freedom and equality in healthcare, GNU solidario, non-gubernamental organization, 2017, http://health.gnu.org/.