Journal of Biomedical Informatics 109 (2020) 103521

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Original Research

A novel gateway-based solution for remote elderly monitoring

Issam Boukhennoufa [a], Abbes Amira [b], Faycal Bensaali [a,*], Sahar Soheilian Esfahani [a]

[a] College of Engineering, Qatar University, Doha, Qatar
[b] Institute of Artificial Intelligence, De Montfort University, Leicester, United Kingdom

ARTICLE INFO

Keywords:
Internet of Things
Remote monitoring
Healthcare
Assisted living
Graphical user interfaces
Clouds

ABSTRACT

Internet of Things (IoT) technologies have been applied to various fields such as manufacturing, automobile industry and healthcare. IoT-based healthcare has a significant impact on real-time remote monitoring of patients' health and consequently improving treatments and reducing healthcare costs. In fact, IoT has made healthcare more reliable, efficient, and accessible. Two major drawbacks which IoT suffers from can be expressed as: first, the limited battery capacity of the sensors is quickly depleted due to the continuous stream of data; second, the dependence of the system on the cloud for computations and processing causes latency in data transmission which is not accepted in real-time monitoring applications.

This research is conducted to develop a real-time, secure, and energy-efficient platform which provides a solution for reducing computation load on the cloud and diminishing data transmission delay. In the proposed platform, the sensors utilize a state-of-the-art power saving technique known as Compressive Sensing (CS). CS allows sensors to retrieve the sensed data using fewer measurements by sending a compressed signal. In this framework, the signal reconstruction and processing are computed locally on a Heterogeneous Multicore Platform (HMP) device to decrease the dependency on the cloud. In addition, a framework has been implemented to control the system, set different parameters, display the data as well as send live notifications to medical experts through the cloud in order to alert them of any eventual hazardous event or abnormality and allow quick interventions. Finally, a case study of the system is presented demonstrating the acquisition and monitoring of the data for a given subject in real-time. The obtained results reveal that the proposed solution reduces 15.4% of energy consumption in sensors, that makes this prototype a good candidate for IoT employment in healthcare.

1. Introduction

IoT-based real-time remote monitoring has been applied in numerous areas such as healthcare, which has witnessed a large growth of interest in recent years [1–3]. Connecting smart devices, machines and humans in IoT ensures the development of efficient healthcare and medical asset monitoring systems. For example, using various types of wearable sensors which can track blood pressure, pulse, red blood cells counts can develop an effective remote health monitoring system capable of sending reports to caregivers and notifying patients to take the medications [4].

IoT-driven health monitoring systems have enabled the clinicians to observe and treat the patients remotely and provided a quick and cost-efficient approach for connected healthcare [5]. The integration of IoT in healthcare has revolutionized the models in terms of efficiency, time, investments and privacy [4]. The continuous stream of data makes the sensors vulnerable to energy consumption problems since they are, in most cases, battery-driven. Furthermore, these systems are completely dependent on the cloud to synthesize and analyze the data yielding to latency issues.

This paper aims at addressing these issues by developing a platform which collects health data remotely from elderly patients and monitors their state at any time. The platform can also notify caregivers when a fall is detected, allowing them to deploy fast and efficient interventions.

In our platform, the monitoring system consists of three parts as in regular IoT designs: wearable medical sensors with embedded Compressive Sensing (CS) for measuring and transmitting relevant information to a close-by gateway which is the primary part of this framework. The gateway is a Heterogenous Multicore Platform (HMP) that oversees data processing incorporating data recovery and other analyses covered in our system, and then sends the processed data to the cloud to be accessed by caregivers. The gateway can store daily activities

* Corresponding author.
E-mail addresses: issam.b@qu.edu.qa (I. Boukhennoufa), abbes.amira@dmu.ac.uk (A. Amira), f.bensaali@qu.edu.qa (F. Bensaali), sahar.esfahani@qu.edu.qa (S. Soheilian Esfahani).

https://doi.org/10.1016/j.jbi.2020.103521
Received 24 October 2019; Received in revised form 7 July 2020; Accepted 24 July 2020
Available online 31 July 2020

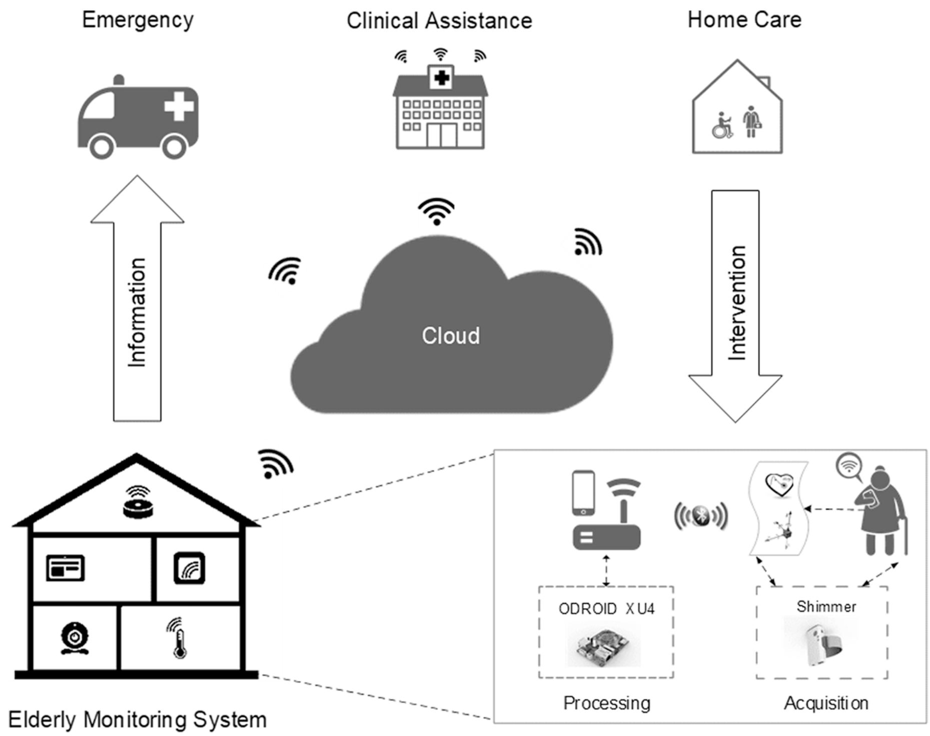

Fig. 1. Proposed IoT system.

and transmit reports periodically or at the end-user request.

The proposed system monitors ElectroCardioGram (ECG) and kinematic data, which can be used to diagnose any kinds of arrhythmias or abnormalities. Besides, ECG biometric recognition has also been integrated into the platform to help identify the patient under supervision. Depending on the scenario, our platform enables the user to set different parameters to economize battery lifespan and control the processing time. This can be performed through a framework or visualization platform, which has been developed to provide user-friendly access to the system. The caregivers and authorized persons can have access to the data through the cloud and receive notifications instantaneously permitting a pre-diagnosis and thus saving treatment time.

The rest of the paper is organized as follows. Related work on IoT-based healthcare, CS and fall detection are presented in Section 2. The proposed system and its different parts are described in Section 3, and Section 4 is concerned with the software implementation of the system. Finally, a case study to validate our framework is reported in Section 5 followed by conclusions in Section 6.

2. Related work

IoT-driven remote monitoring systems are network-connected sensors continuously gathering and sending data of interest to a nearby gateway. The gateway - commonly a smartphone - transmits the data to a distant IoT object where the captured information can be monitored and examined by the end-user [6]. Due to the growth and evolvement of the data collection and transmission technologies, more IoT-based healthcare applications have been developed in recent years [7]. In [8], the authors discussed the necessity of integrating IoT technology and healthcare solutions. They developed and focused on a continuous monitoring system for patients at risk of high blood pressure to provide them with fast treatment by remote clinical experts. Another effort has been made to employ portable devices and several communication protocols to generate E-health applications [9]. Moreover [10] proposed a model that detects the abnormal interpretations in the ECG and Heart Rate (HR) of a patient, and therefore reduce the critical level of the patient by following precautionary measures at an earlier time. In [11], the authors analyzed the security vulnerabilities and the potential risks that the medical applications detect, it has been classified into remote monitoring, diagnostic support and treatment support. In addition [12,13], the works proposed two techniques to encrypt medical data in order to have secure data transmission for sensitive information.

Recently, the growth of the elderly population derives the attention of more research in IoT healthcare monitoring scope to develop efficient platforms and applications improving healthcare services and reducing the treatment costs for elderlies. For instance, Islam et al. have explored several medical IoT-based applications for remote elderly healthcare follow-up and fitness programs [14]. In addition, many research has been conducted to evaluate the stability and mobility for ageing people; Timed Up and Go (TUG) is one of them [15]. TUG studies the time needed by a subject to complete different segments of movement including getting up from a chair, walking for a 3 m distance, coming back and sitting again. Subjects with higher TUG are said to have a higher risk of falling. However, these studies do not provide information indicating which segment is more indicative of fall risk.

Several efforts have been dedicated to developing an accurate fall detection system using wearable sensors [16]. For instance, to give a quantitative evaluation of walk and turn during the TUG, kinematic sensors have been widely used permitting a more precise and accurate determination of a potential fall [17,18]. Noury et al. have presented an elderly monitoring system by employing kinematic sensors to detect falls and find the most reliable fall scenarios in order to simulate different fall situations [19].

There are also some studies with the aim of analyzing the changes of the acceleration vector over a fall, and coming up with a fall recognition algorithm based on the various stages (i.e. beginning of the fall, shock, aftershock and posture) [20]. In order to improve the results and performance of the previous works, other researchers applied gyroscopes and barometers into their proposed systems [21–23]. Casilari et al. have integrated an accelerometer to a smart watch to recognize a fall. Once a fall is detected, the data will be transmitted via Bluetooth to a smartphone, which serves as a gateway and can send an alert to the cloud [24].

In [25], the authors have presented wearable sensing devices for

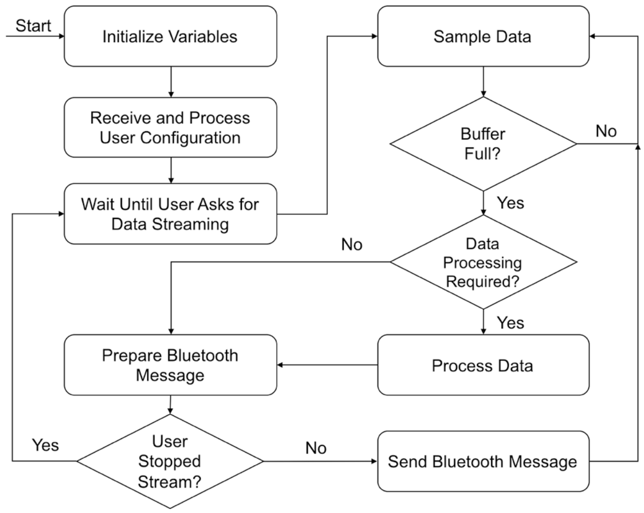

Fig. 2. Different steps of data acquisition and transmission in the modified firmware.

consistent fall recognition. The sensors were a Micro-Electro-Mechanical System (MEMS) based accelerometers transmitting data via Radio Frequency (RF). They have also provided a solution to detect the position of the person. Wu et al. have used a sensor based on Global System for Mobile (GSM) communication and 3D accelerometer to detect and also locate a fall [26].

Despite all the efforts made in IoT-based remote monitoring and fall detection systems, performance of these real-time systems still suffers from the energy efficiency and delay issues. In general, transmission of the sensed data to the cloud for further processing and analyzing will enforce an un-tolerable delay for real-time purposes. Moreover, these systems often operate in a long-term mode which imposes a large amount of data transmission to the power-limited sensing devices and thus, considerable amount of energy is depleted.

To tackle the power consumption issue, applying data compression techniques can be helpful but most of these methods are energy consuming and cannot be employed in IoT platforms. Recently, an emerging signal processing technique called CS has been developed as an alternative method to the Shannon-Nyquist theorem [27]. This method provides a reduction form of the sensed data if the signal is known to be sparse or compressible [28,29]. In fact, CS gives a solution to take a few amount of data at the self-powered sensors and then shift the decoding process to the receiver with no energy constrains. However, the signal reconstruction in CS is more complex in comparison to the traditional compression techniques [30], the results of our previous works reveal that employing CS can reduce the amount of transmitted data and hence, decrease significantly the energy consumption at the sensor level [31,32].

There are some research exploring CS in Wireless Sensor Networks (WSN), which operates in long-term data acquisition mode [33–35]. Dixon et al. have used CS with ECG data and compressed ECG by almost 16 times by changing sensing matrix and recovery processes [36]. Another study has been conducted to demonstrate the preeminence of CS over the wavelet-based ECG compression using the Shimmer™ sensing device [37]. It is shown that although the wavelet compression performs better in terms of quality of the reconstructed signal, CS can improve the power consumption of the sensor by 9.7%. In this study, noiseless MIT ECG records have been used as the sensor input and thus, optimal signal compression has been achieved. Furthermore, they have performed CS data recovery remotely by the SPGL1 solver [38]. Besides the aforementioned results, there are more studies conducted to prove the improvements and advantages of CS and its ability to be employed in IoT applications [39,40]. Other compression techniques are also available, in Gurve et al. [41] reported 11-fold compression with very small reconstruction error in a lossless algorithm suitable for use on cell telephones.

Table 1
Shimmer commands.

Command	Description
SET_SAMPLING_RATE	Sets the Shimmer sampling frequency
SET_SENSORS	Activates/deactivates the Shimmer sensor. It takes three bytes: 1. First byte is 0x00 or 0x80 for OFF or Accelerometer 2. Second byte is 0x00 or 0x20 for OFF or Battery Sensor 3. Third byte is 0x00 or 0x10 for OFF or ECG Senor
SHIMMER_COMMANDS	Configures the ECG parameters such as gain. It takes 10 bytes which can be copied from the Shimmer CONSENSYS Software
SET_WINDOW_AND_MODE	Sets the required buffer size and processing modes. The command takes three bytes: 1. First byte is 0x01, 0x02, or 0x03 for real-time transmission, 1-second buffer, and 1.5-seconds buffer. 2. Second byte defines the acceleration processing mode and can be 0x00, 0x01, 0x02, or 0x03 for OFF, RAW (no processing), ADJUSTED (acceleration samples are placed in 14 bits instead of 16 bits per sample), or LQ COMPRESS (low quality compression) 3. The third byte defines the ECG processing mode and can be 0x00, 0x01, 0x02, 0x03, or 0x04 for OFF, RAW, ADJUSTED (8 bits per sample instead of 16 bits), HQ COMPRESS (high quality compression suitable for clinical monitoring) or LQ COMPRESS (low quality compression only suitable for data analysis)
START_STREAMING_COMMAND	Starts the Shimmer streaming
STOP_STREAMING_COMMAND	Stops the Shimmer streaming

As mentioned earlier, delay is one of the most important challenges in the IoT health monitoring systems, imposed to the system by transmitting the sensed data to a cloud for further processing and analyzing. In [42], the authors have revealed the good performance of real-time healthcare services and urged the need for edge computing by developing a real-time IoT-based gateway monitoring system. Recently, HMP has been employed for various processing and analyzing tasks to shift the computation load from the cloud. It is shown that by exploiting different cores, the processing time can be significantly decreased [43]. In [44], the authors succeeded to reconstruct ECG data in real-time on an apple smartphone. They used an adjusted form of the Iterative Shrinkage Thresholding Algorithm (ISTA) but didn't take into account tradeoffs between the important parameters such as energy consumption, processing time and dimensions of the signal. In another study, Pareschi et al. have presented a solution for live energy efficient ECG data decoding developed on an ARM's Cortex-M4F-microcontroller [45]. They compressed the data using a modified rakeness-based sensing matrix and hence, could reduce the reconstruction complexity compared to the regular sensing matrices.

This paper presents an IoT-based solution which acquires ECG and kinematic data through a CS equipped sensor, processes them on a local HMP and notifies medical experts through the cloud. A set of comprehensive tests using data from the MIT database and Shimmer sensor device have also been carried out to evaluate the performance of the system.

The main contributions of this work can be summarized as follows:

- Signal level: ECG and kinematic data compression performances are evaluated by means of signals obtained from the sensor.

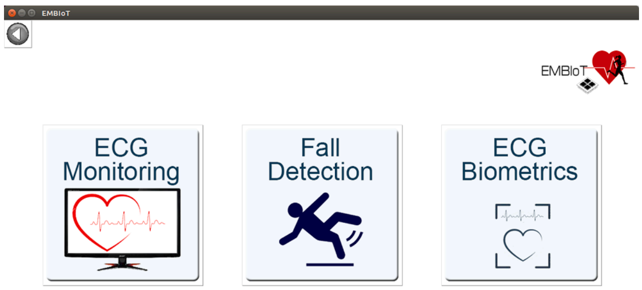

Fig. 3. The main interface of the framework.

- Sensor level: the firmware of the ShimmerTM wearable device is modified to send either the actual data or the CS compressed data at the user convenience.
- Gateway level: the data reconstruction is performed on a local HMP device using two different algorithms; Subspace Pursuit (SP) and Orthogonal Matching Pursuit (OMP). Moreover, identifying the subject by using ECG biometric as well as sending a fall alert through the cloud, in case of the event occurrence, are implemented on the gateway.

3. System overview

In our IoT-driven real-time health monitoring system patients are monitored in real-time permitting the caregivers to access their data at anytime as discussed in [46]. It consists of three parts: wearable sensor, gateway and the cloud. The overview of the system is represented in Fig. 1. In this system, we have used Shimmer as the sensing device which sends data of interest to a close-by gateway. The gateway is an ODROID XU4 board which handles various data processing such as data reconstruction, biometric recognition and fall detection. The resulting information is then transmitted to the clinical experts through the cloud.

In the following sub-sections, a description of each building block of the proposed system is given, namely the wearable sensor, the gateway and the mobile App.

3.1. The wearable sensing device

The Shimmer3TM ECG/EMG unit is a state-of-the-art sensor able to stream ECG and kinematic data simultaneously. On the hardware level, it uses a 16 Kb RAM – 16 bit – 24 MHz microcontroller supplied by Texas Instruments [47]. Software-wise, the firmware is called LogAndStream and can be modified and compiled through Code Composer Studio. It is written in C programming language and hosted on GitHub repository [48]. The modified firmware can be uploaded to the Shimmer sensor using CONSENSYS software provided by the supplier. The ECG unit of the sensor can be set up to record skin electrical signals including ECG. In fact, measuring the pathway of the electrical data going to the heart can be set up by three-lead body connections. These data can be measured during different states of the patient such as exercise and rest, and thus, provide information on the heart status during physical exertion [49]. Also, the unit includes a kinematic sensor providing a 3D acceleration data vector used in the proposed platform to detect a fall.

3.2. The ODROID IoT device

A hardkernel ODROID XU4 board featuring ARM's big.LITTLE heterogeneous octa-core solution is used as the gateway of the proposed system. It runs on Ubuntu or Android and consists of 8 cores; 4 big A15 cores and 4 LITTLE A7 cores [50]. The reason for selecting this device is that its configuration can be found on most of today's smartphones and therefore, it is a good candidate for evaluating the system.

This gateway is capable of saving daily activities of the patient including patterns of the different physical or potential physiological observed parameters, in order to be investigated over time. This platform enables the end-user to request for instant or periodic reports. Moreover, caregivers can be informed of any potentially hazardous event by the event emergency (i.e. fall) classification performed at the gateway. Once an emergency is reported, the gateway will transmit physiological information continuously for early diagnosis. As stated before, this methodology reduces system latency and improves the stability issues, imposed by the dependence of computations on the cloud [46] and hence, eliminates the extra burden on the cloud.

Moreover, the ODROID-VU7 is used to visualize the data. It is a 7-inch multi-touch screen for ODROIDs that gives users the ability to create all-in-one integrated projects such as tablets, game consoles, infotainment systems and embedded systems. The 800 × 480 display connects to ODROID via an HDMI link board and a micro-USB link board which handles power and signal. This high-quality touchscreen is specifically designed to work with both Android and Linux on the ODROID-XU4. Our system can also be implemented in a tablet-like form using this ODROID multi-touch screen.

Table 2
Operational mode scenarios.

Scenario\Parameter	Transmission Type	Security	Core	Frequency
Regular Transmission	Regular transmission	Off	Small core	0.8 GHz
Minimum Consumption	CS-based transmission	Off	Small core	0.8 GHz
Optimal Consumption	CS-based transmission	Off	Big core	1.2 GHz
Secured Data	CS-based transmission	On	Big core	1.2 GHz

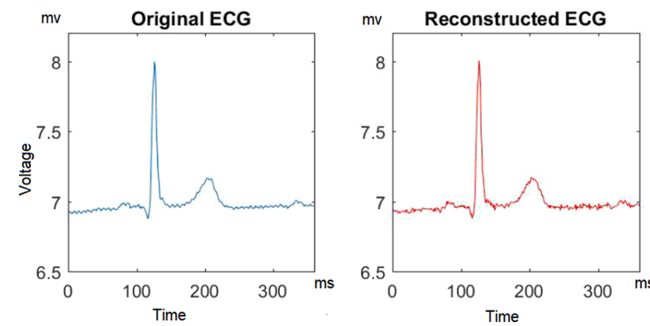

Fig. 4. Original and reconstructed ECG signal from the Shimmer device.

3.3. Mobile application

This subsection is concerned with the developed application, called Vitals Monitoring, which runs on Android devices. The role of this app revolves mainly around data visualization for both elderlies and caregivers. It also delivers an alarming feature for caregivers in case of a fall event. Regarding the functionality, when a user logs in, the application will behave depending on the category the user belongs to. When a caregiver signs-in, a list of elderlies will appear. The application enables the caregiver to observe details of the patients, visualize the streamed data received from the elderlies or even call them for a quick check-up. Otherwise, if the logged-in user is an elderly, he/she will gain the ability to visualize the data streamed from the Shimmer3ECG connected to him/her and ask for help when needed. More details about the features of the application and the different interfaces that the patients and caregivers can have access to, are presented in Section 5.

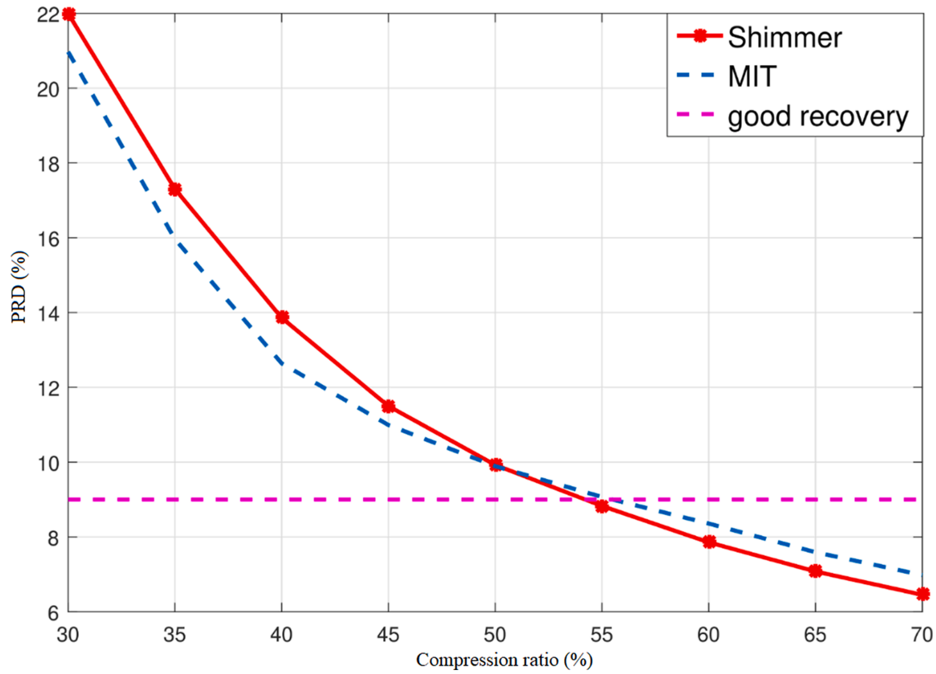

Fig. 5. Reconstruction quality in terms of PRD.

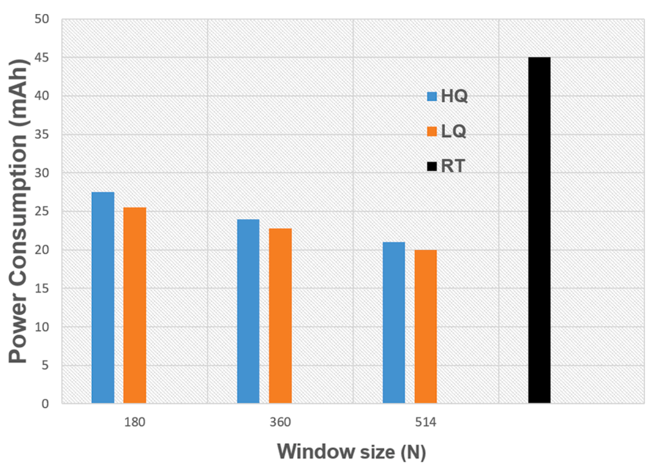

Fig. 6. The Shimmer's power consumption in real-time (RT), i,e. data transmitted.

4. Software implementation

In this section, software implementation of different parts of the system are described namely: acquisition and processing of data, visualization framework and the cloud programming

4.1. Acquisition and processing

On the original firmware, the Shimmer sensor streams ECG and acceleration data at 512 Hz and 60 Hz sampling rate, respectively. The continuous flow of data makes the Shimmer vulnerable to power consumption. To remedy to this and give the user control over the device's power consumption, the original firmware has been customized to perform CS and is discussed in details in our previous works [31,32]. Fig. 2 illustrates the steps of sensing and transmitting data (compressed or raw) using the modified firmware.

As discussed earlier, all the processing and analysis are carried out on the ODROID device. If the received data at the gateway are raw, no further processing is required and data will be sent directly to the framework to be displayed. Otherwise, when the received data are compressed, signal reconstruction will be carried out on the gateway. The user can select between two decoding algorithms available in the framework (SP, OMP).

Data reconstruction in the proposed platform by employing SP or OMP algorithm is discussed in details in our previous works [46,43]. Besides, the gateway implements some data analytics algorithms on the reconstructed data in order to provide more precise information for early diagnosis. When the platform is set to monitor ECG, the subject can be identified through the ECG biometrics as well. In case of fall detection, the medical experts will be notified instantaneously through a cloud alert received on their connected device.

The processing performed on the gateway was programmed using C++ and python programming languages. The main python program is called *Embiot.py* and contains a visualization platform. It is considered as the parent application which owns upper-level commands for its children applications. The parent can generate autonomous child processes responsible for getting data from the Shimmer. Each child process can also produce its own C++ processes to do complicated data computations such as CS recovery and data analysis. The operation of the parent is based on three classes; the *Shimmer* class, the *Child_Manager* class and the *Bluetooth_Manager* class. The latter handles different Bluetooth operations such as enabling the application to scan, pair or connect to the Bluetooth devices around. It can also filter Bluetooth devices based on their address and find the available Shimmers. The *Shimmer* class configures Shimmer devices for a specific data acquisition and sets different parameters such as CS, window length, ECG or Acceleration Table 1 summarizes the Shimmer configuration commands introduced in the modified firmware. Finally, the *Child_Manager* method handles communication between various processes. All the data reconstruction and analytics algorithms are performed in this class.

Additionally, different libraries have been used in the proposed platform such as Armadillo for linear algebra and scientific computing in order to implement matrix operations [51], and ZeroMQ library for communication between python and C++ through InterProcess-Communication (IPC) socket [52]. A common data serializer called Google's Protocol Buffer has also been used due to the ZMQ limitation on using serialized data and strings. This data serializer is more

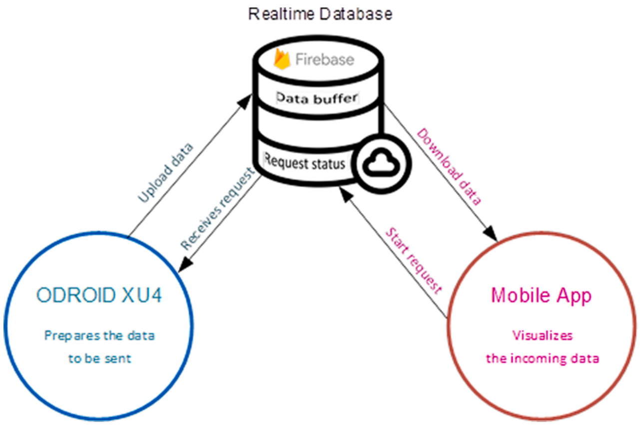

Fig. 7. Request-Response communication with Firebase.

and send notifications to alert the user of pertinent events.

The framework operates on two different modes: analysis mode and operational mode. In the analysis mode, the user manually sets the parameters which are divided into two types: acquisition and processing parameters. Acquisition parameters are Shimmer associated parameters and allow the user to set regular or CS data acquisition, the length of the transmitted window (1 s or 1.5 s), and type of data security. Processing parameters are those related to ODROID and permit to select the reconstruction algorithm, in case of CS-based transmission, and the number of cores and frequencies to be employed in computations. In operational mode, the parameters are set via a pre-defined scenario, which can be selected from a drop-down menu. Once the scenario is selected, parameters will be set automatically and data monitoring starts

Different scenarios of the operational mode are listed in Table 2 Operational mode scenarios. These scenarios have been discussed in details in our previous work [46].

Fig. 4 shows A plot of both the original streamed ECG data on the

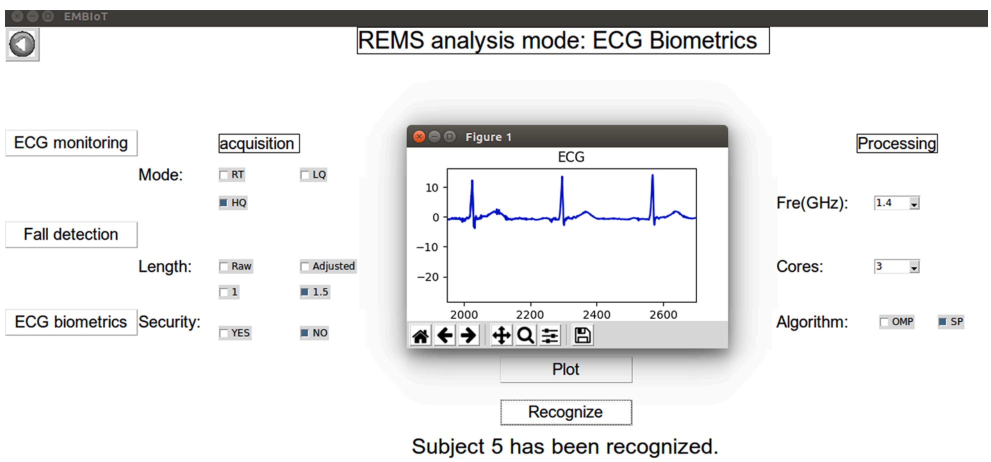

Fig. 8. ECG biometric interface.

convenient for real-time applications since it requires less computation in comparison to the conventional JavaScript Object Notation (JSON) data serializer [53].

4.2. Visualization framework

A visual framework has been developed using Python Tkinter module in order to enable the end-users and clinical experts to easily access the platform and configure the system. It allows the users to connect the platform to the available nearby Shimmer sensors using Bluetooth. The users can activate the sensors (ECG or kinematic) and configure them for the appropriate data acquisition algorithm (regular or CS-based) via the framework. Also, the user can manage the data security; transmission of data can be either secure, in case of confidential information, or insecure if confidentiality is not required. The framework also permits to control data processing on the gateway; this means the user can select an appropriate algorithm for data reconstruction (SP or OMP). In the case of CS-based data acquisition, the clinical expert can decide on the type and number of cores involved for processing as well as their frequencies and hence, manage the processing time. Moreover, the framework provides different data analytics including biometric recognition for ECG and fall detection for the kinematic sensor. Once the system is correctly configured, the acquisition of data starts and the interface displays the data in question. In case of a fall, a green flashlight will be displayed on the interface Fig. 3. The main interface of the framework. shows the main interface of the framework. This framework also allows communication with the cloud. It permits to visualize the data on a remote smartphone equipped with our developed Android application

shimmer and the reconstructed ECG signal decoded in our framework.

Besides, results showed in Fig. 5 showed that the reconstruction quality improves greatly when the Compression Ratio (CR) increases. In this paper, *CR* is defined to be the percentage ratio of the number of the compressed signal elements by the number of elements in the original signal i.e.$CR = 100{*}\frac{m}{n}$. The reconstruction quality of data is quantified using the percentage root-mean-square difference (PRD): $PRD = \frac{\|X-\hat{X}\|}{\|X-\mu\|}$ where μ denotes the mean of the signal.

Besides, a reasonable reconstruction quality is achieved when *CR* equals 50% and 55% in which the obtained PRD is 9 for both the MIT database and for the Shimmer device respectively. The MIT database shows a better recovery performance when compared to the Shimmer due to the presence of the noise level in the data of the latter. Moreover, it is legitimate to mention that for the case where MIT database is used, the results were averaged over all the records belonging to the data set. It is worth noting also that when using all the records in the MIT database, the obtained result shows to be inferior to the results reported in [54]. For instance, taking the case of $CR = 0.25$, a $PRD = 9$ and $PRD = 12$ has been achieved by this work herein and the work in [54] respectively.

To further evaluate the performance of the proposed framework, a digital-based CS compression has been implemented on the Shimmer sensing node. Subsequently, an energy consumption analysis has been carried out to quantify the performance of deploying a sparse sensing matrix for a CS-based ECG compression.

The implementation is performed over three different ECG segments window sizes N = {180, 360, 514}.

The Shimmer's power consumption in Real-Time (RT) (data

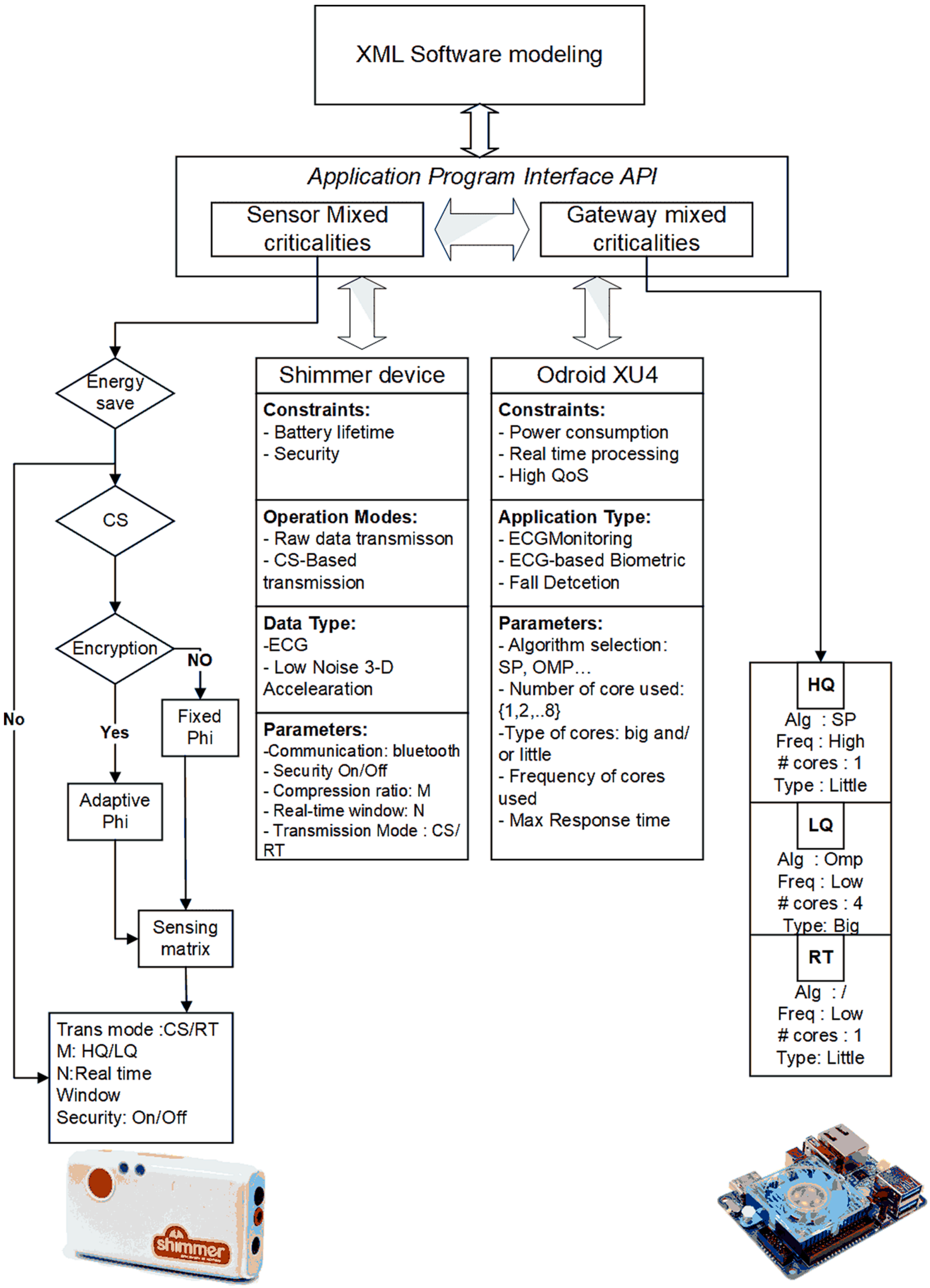

Fig. 9. Summary of the system's functionalities.

Table 3
Collected personal information from the subjects.

Subject #	Age (years)	Height (cm)	Weight (Kg)
Subject 1	21	172	65
Subject 2	22	183	100
Subject 3	21	186	66
Subject 4	22	188	72
Subject 5	22	189	98
Subject 6	21	187	94
Subject 7	21	190	130

transmitted without compression), HQ and LQ scenarios are shown in Fig. 6.

It shows that data directly from the Shimmer without compression is reduced by a factor of 15%. Additionally, the results obtained also show that using a window of 1.5 sec would reduce power consumption by a factor of approximately 1.5.

In our work, we set up two scenarios for data recovery, high quality (HQ) and low quality (LQ). In HQ the reconstructed ECG data is highly correlated with the original signal and presents minimal errors and can be directly used for human-based diagnosis (PRD < 7), while in the LQ

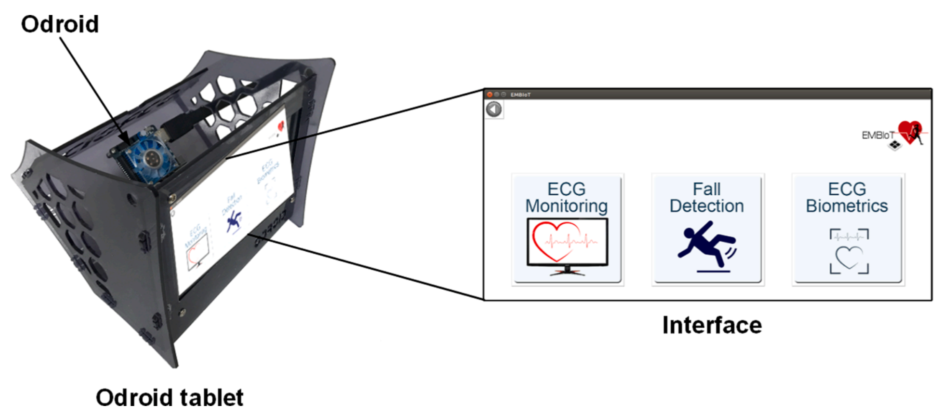

Fig. 10. System runs on ODROID IoT device with touch screen.

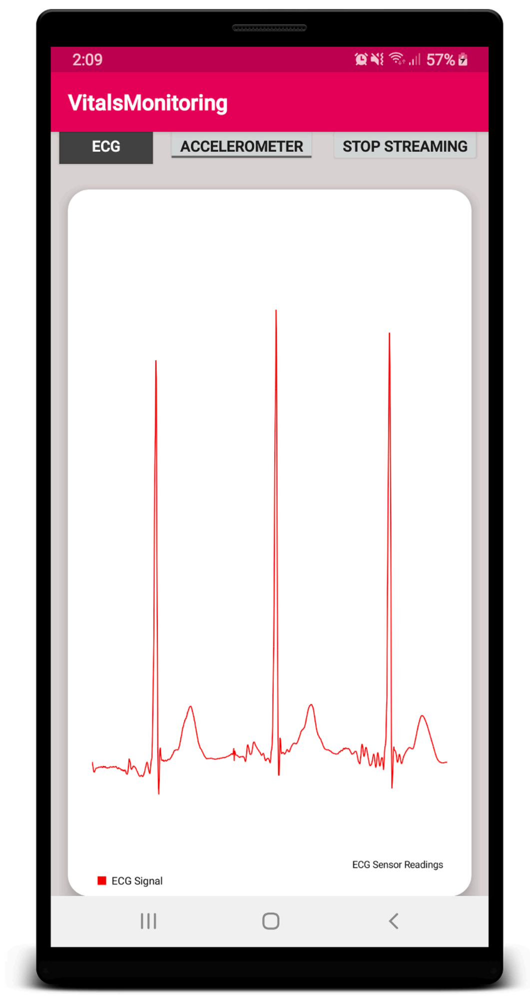

Fig. 11. Mobile application interface.

setting the ECG data quality is prone to degradation but still its features can be detected and employed for computer-based classification ($7 < \text{PRD} < 18$) [46].

4.3. Communication with the cloud

In this section, the setup of the cloud-hosted database platform is described as well as the role of the HMP and mobile application on data visualization streamed through the cloud. In the proposed platform, the database is meant to act as a temporary buffer of the uploaded data in which the data can be accessed remotely by other clients with the correct authenticated privilege. For example, a mobile application which can display the stream of accelerometer and/or ECG data. In our approach, the stored data is continually updated with the new values, while the mobile application "listens" to any changes in the data and handles them accordingly. To achieve this, we have employed Firebase as the system database on the cloud.

Firebase platform offers a cloud-hosted database system called the Realtime Database that follows the NoSQL mechanism which is simpler and less structured than the traditional relational database. This allows the real-time storage of large data sets which can be synchronized to all the clients connected to the database [55].

In this platform, the communication with the Firebase Realtime Database is performed on the ODROID XU4. To start the interaction with the cloud server, an account must be created on Realtime Database, then a Firebase 'project' can be set up. In order to implement data using the REST API, an appropriate helper library compatible with the programming language must be installed. In this case, a separate Python program would be run by the HMP which acts as a client device and stands by for the data until streaming is required. Therefore, the Pyrebase helper library for Python is installed on the ODROID XU4 to handle the communication with the cloud database.

Once all the setup requirements have been met, the process of uploading and storing data in database can be started. A Python program is developed to upload and store the received data pool into an array on database. By using Pyrebase, the 'update' and 'set' functions are applied to establish a proper connection with the cloud database through the internet and then a single HTTP PUT request has been used to send the data. As a result, synchronization happens when the data, which are put into the cloud database, are in accordance with the database referenced from the Python program.

As discussed earlier, the data can also be visualized on a remote Android device connected to internet. In our proposed mobile application (Vitals Monitoring), ECG streaming is achieved once a flag called Stream_Requests on the database is incremented. ODROID XU4 is continuously listening to the value of this flag that is specific for the appropriate elderly. Once the flag value is incremented, the streaming will start and then the ODROID XU4 will populate the ECG_Data branches for that elderly simultaneously. Consequently, multiple users have the ability to visualize real-time data at the same time. When the user (caregiver or elderly) no longer requests for the data, he/she will press the *Stop Streaming* button and therefore, decrement the value of Stream_Requests flag by one. This can also be done automatically when the application is not running in the foreground (i.e. it is running in the background or closed). ODROID XU4 will continue listening to the value of the Stream_Requests flag as long as it reaches zero which means no one is listening and there is no need for data streaming anymore. This

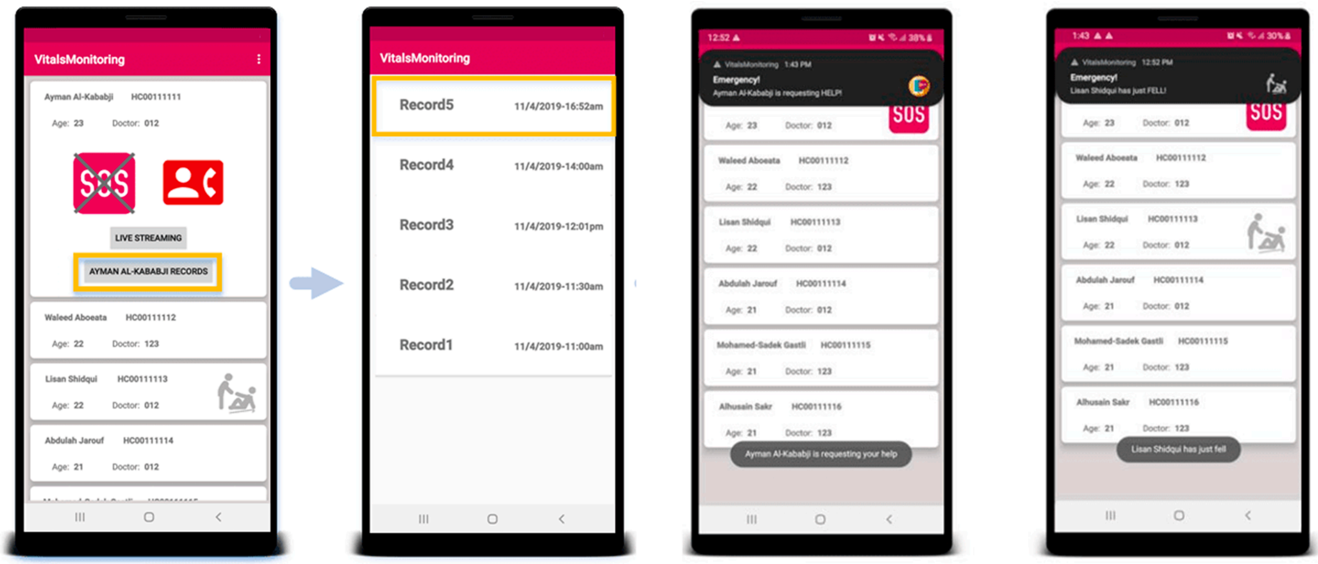

Fig. 12. Different feature of vitals monitoring application; (a) checking previous fall records of a specific patient; (b) help request sent by an elderly; (c) a fall is detected.

achieves a more optimized streaming system. Fig. 7 illustrates the data live streaming process through the Firebase.

5. Case study

To illustrate different features of our system, a case study is presented here, which is based on an ECG acquisition followed by a biometric recognition. The visual framework encompasses the configuration of the system and the streaming of data, which will also be shown on the mobile app:

When the framework is running, it will scan the available Bluetooth devices around and select the available Shimmer sensors by using Ubuntu's Bluetoothctl module [56]. The system pairs automatically with the Shimmer using the supplier provided password which is the same for all the Shimmer sensors. Because ECG monitoring was examined in this case study, the leads should be placed in 3-lead connections on the subject's chest.

Once the setup is ready, the user can configure different parameters through the framework. Fig. 8 depicts the framework interface for the ECG biometrics. A simple click on the buttons on the left side of the interface allows switching to another application (fall detection or ECG monitoring).

It is worth mentioning that there are two types of configurable parameters; acquisition and processing. Acquisition parameters (shown at the left side of the interface) are *Mode* to activate CS-based transmission or not, *Length* to select the size of the window for signal reconstruction, and *Security* to be applied in case of confidential data. Processing parameters (shown at the right side of the interface) includes *Fre* to set the frequency of the ODROID CPUs which ranges from 800 MHz to 2000 MHz, *Cores* to select the number of cores involved in processing, and *Algorithm* to select the appropriate decoding algorithm (OMP or SP). Fig. 9 summarizes the above-mentioned properties of the system.

In this study, CS-based data acquisition by selecting HQ for the compression rate and 1.5 s window is performed. 1.5 s window indicates that the compression is done every 540 values (ECG frequency is 360 Hz and 540 corresponds to 1.5 s of data). In this transmission, security is not applied. For the processing parameters, 3 cores are selected to operate with 1.4 GHz frequency. The reconstruction algorithm is also set to SP. When the setup of the parameters is completed, a click on the *plot* button starts ECG streaming and live displaying of the data.

In order to plot the data, matplotlib module from Python was used. Whereas ECG frequency was 360 Hz, the highest frequency in matplotlib for displaying real-time data was 100 Hz. This condition imposed undesired latency to the system that caused the loss of the live streaming aspect. To fix this, we stored the reconstructed data in a text file for each streaming window and when the last value of the data was received, the complete period was plotted.

A click on the *recognize* button displays the number related to the identified subject. For this purpose, a database is already created using several ECG samples from various subjects. Thus, the sensed ECG is compared to this database and the closest sample is picked up as the recognized subject [57,58]. Table 3 represents the age, height and weight of some subjects sampled by Shimmer at frequencies of 512 Hz and approximate capturing duration of one minute. In this study, subject 5 is recognized (see Fig. 8).

Fig. 10 displays the ODROID IoT device running the framework and visualizing the results for subject 5.

The real-time results of monitoring this subject on our developed Vitals Monitoring application is illustrated in Fig. 11.

There are more features available in this application helping the elderlies and caregivers to interact with each other and monitor the current status of the elderly easier. Fig. 12 represents different interfaces which a caregiver has access to (a) view the history of a specific patient's fall. This helps the caregiver check all the previously stored data of the patient for further analysis; (b) be alerted when a help request is sent; (c) be notified when an elderly falls. Two latter features ensure a quick and real-time response from the caregivers.

6. Conclusion

In this paper, we have presented an IoT-based remote elderly monitoring system that addresses the main issues similar platforms are dealing with; power consumption in the sensors and network latency. In order to increase the lifespan of the wearable medical devices, CS a state of the art power saving technique has been implemented on the sensing node. The absolute dependency of the processing on the cloud is tackled by introducing a local HMP gateway to reduce the burden on the cloud and therefore, make the platform faster and more customizable. In addition, a framework is developed to access and visualize data through a user interface. The data can also be accessed through our developed mobile application on an Android device connected to the cloud. In this proposed platform, alerts can be sent to the clinical experts to deploy faster interventions. A case study has been reported to illustrate different functionalities of the system by taking a patient's ECG, monitoring it and identifying the subject via ECG biometrics. The information has also been transmitted to a remote Android device.

As future work, the proposed system could be further developed to integrate into other IoT application in the healthcare area such as robotic surgery and smart ambulance systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper was made possible by the National Priorities Research Program (NPRP) grant No. 9-114-2-055 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] A.M. Ghosh, D. Halder, S.A. Hossain, Remote health monitoring system through IoT, in: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 2016.
[2] O. Sadiku, K. Eze, M. Musa, Wireless sensor networks for healthcare, J. Sci. Eng. Res., vol. 5, n° %17, 2018, pp. 210–213.
[3] M. Shoaib, H. Garudadri, Digital pacer detection in diagnostic grade, 2011 IEEE 13th International Conference on e-Health Networking, Applications and Services, Country, United States, 2011.
[4] S. Tyagi, A. Agarwal, P. Maheshwari, A conceptual framework for IoT-based healthcare system using cloud computing, in: 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), Noida, India, 2016.
[5] M.S. Jassas, A.A. Qasem, Q.H. Mahmoud, A smart system connecting e-health sensors and the cloud, in: 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, Canada, 2015.
[6] D. Minoli, Building the Internet of Things with ipv6 and mipv6, Hoboken, John Wiley & Sons Inc, New Jersey, 2013.
[7] K. Rose, S. Eldridge, L. Chapin, The internet of things: An overview, The Internet Society (ISOC) 80 (2015) 1–50.
[8] T. Takpor, A.A. Atayero, Integrating Internet of Things and EHealth solutions for students' healthcare. Proceedings of the World Congress on Engineering, 2015.
[9] G. Sebestyen, A. Hangan, S. Oniga, Z. Gal, eHealth solutions in the context of Internet of Things, 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 2014.
[10] S. Palanivel Rajan, S. Rajamony, Viable investigations and real-time recitation of enhanced ECG-based cardiac telemonitoring system for homecare applications: a systematic evaluation, TELEMEDICINE and e-HEALTH , April 2013.
[11] Ahmed Abdelaziz, Mohamed Elhoseny, Ahmed S. Salama, A.M. Riad, A machine learning model for improving healthcare services on cloud computing environment, Measurement 119 (2018) 117–128, https://doi.org/10.1016/j.measurement.2018.01.022.
[12] M. Elhoseny, G. Ramírez, O. Abu-Elnasr, Secure medical data transmission model for IoT-based healthcare systems, IEEE Access (2019) 1–15.
[13] Hamza Djelouat, Abbes Amira, Faycal Bensaali, Issam Boukhennoufa, Secure compressive sensing for ECG monitoring, Comput. Security 88 (2020) 101649, https://doi.org/10.1016/j.cose.2019.101649.
[14] S.R. Islam, D. Kwak, M.H. Kabir, M. Hossain, K.-S. Kwak, The internet of things for health care: A comprehensive survey, IEEE Access 3 (2015) 678–708.
[15] G.A. Meijer, K.R. Westerterp, F.M. Verhoeven, H.B. Koper, F. ten Hoor, Methods to assess physical activity with special reference to motion sensors and accelerometers, IEEE Trans. Biomed. Eng. vol. 38, n° %13 (1991) 221–229.
[16] T.N. Gia, V.K. Sarker, I. Tcarenko, A.M. Rahmani, T. Westerlund, P. Liljeberg et H. Tenhunen,, Energy efficient wearable sensor node for IoT-based fall detection systems, Microprocess. Microsyst. 56 (2018) 34–46.
[17] Y. Higashi, K. Yamakoshi, T. Fujimoto, M. Sekine et T. Tamura,, Quantitative evaluation of movement using the timed up-and-go test, n° %14, IEEE Eng. Med. Biol. 27 (2008) 38–46.
[18] T. Tmaura, N.A. Zakaria, Y. Kuwae, M. Sekine, K. Minato, M. Yoshida, Quantitative analysis of the fall-risk assessment test with wearable inertia sensors, 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Japan, Osaka, 2013.
[19] N. Noury, P. Rumeau, A. Bourke, G. 'OLaighin, J. Lundy, A proposal for the classification and evaluation of fall detectors, n° %16, IRBM 29 (2008) 340–349.
[20] U. Lindemann, A. Hock, M. Stuber, W. Keck, C. Becker, Evaluation of a fall detector based on accelerometers: A pilot study, n° %14, Med. Biol. Eng. Compu. 43 (2005) 548–551.
[21] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, S. Valenti, A high reliability wearable device for elderly fall detection, IEEE Sensors J. (2015) 4544–4553. Vols. %1 sur %215,, n° %15.
[22] C. Wang, W. Lu, M.R. Narayanan, D.C.W. Chang, S.R. Lord, S.J. Redmond, N. H. Lovell, A low-power fall detector using triaxial accelerometry and barometric pressure sensing, n° %16, IEEE Trans. Ind. Inf. 12 (2016) 2302–2311.
[23] A. Bourke, K. O'donovan, G. Olaighin, The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls, n° %17, Med. Eng. Phys. 30 (2008) 937–946.
[24] E. Casilari, M.A. Oviedo-Jim'enez, Automatic fall detection system based on the combined use of a smartphone and a smartwatch, n° %111, PLoS ONE 10 (2015) e0140929.
[25] D. Chen, W. Feng, Y. Zhang, X. li, A wearable wireless fall detection system with accelerators, 2011 IEEE International Conference on Robotics and Biomimetics, Phuket, Thailand, 2011.
[26] F. Wu, H. Zhao, Y. Zhao, H. Zhong, Development of a Wearable-Sensor-Based fall detection system, Int. J. Telemedicine Appl. 2015 (2015) 2.
[27] E.J. Candès, R. Justin, T. Terence, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, n° %12, IEEE Trans. Inf. Theory 52 (2006) 489–509.
[28] E. Candes, T. Tao, Decoding by linear programming, n° %112, IEEE Traactions on Information Theory 51 (2005) 4203–4215.
[29] D.L. Donoho, Compressed senssing, IEEE Trans. Inf. Theory (2006) 1289–1306.
[30] M.M. Baig, H. Gholamhosseini, M.J. Connolly, A comprehensive survey of wearable and wireless ECG monitoring systems for older adults, n° %15, Med. Biol. Eng. Compu. 51 (2013) 485–495.
[31] H. Djelouat, A. Amira, F. Bensaali, Compressive sensing-based IoT applications: A review, n° %14, J. Sensor Actuator Networks 7 (2018) 45.
[32] H. Djelouat, A. Amira, F. Bensaali, C. Kotronis, E. Politi, M. Nikolaidou, G. Dimitrakopoulos, Real-time ECG monitoring using compressive sensing on a heterogeneous multicore edge-device, Microprocess. Microsyst. (2019).
[33] S. Li, L. Da Xu, X. Wang, Compressed sensing signal and data acquisition in wireless sensor networks and internet of things, n° %14, IEEE Trans. Ind. Inf. 9 (2012) 2177–2186.
[34] C. Karakus, A.C. Gurbuz, B. Tavli, Analysis of energy efficiency of compressive sensing in wireless sensor networks, n° %15, IEEE Sens. J. 13 (2013) 1999–2008.
[35] F. Fazel, M. Fazel, M. Stojanovic, Random access compressed sensing for energy-efficient underwater sensor networks, n° %18, IEEE J. Sel. Areas Commun. 29 (2011) 1660–1670.
[36] A. Dixon, E. Allstot, D. Gangopadhyay, Compressed sensing system considerations for ECG and EMG wireless biosensors, IEEE Trans. Biomed. Circuits Syst. 6 (2012) 156–166.
[37] H. Mamaghanian, N. Khaled, D. Atienza, P. Vandergheynst, Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes, n° %19, IEEE Trans. Biomed. Eng. 58 (2011) 2456–2466.
[38] E. Van Den Berg, M. Friedlander, Probing the Pareto frontier for basis pursuit solutions, n° %12, SIAM J. Sci. Comput. 31 (2008) 890–912.
[39] F. Abate, V.K. Huang, G. Monte, V. Paciello, A. Pietrosanto, A comparison between sensor signal preprocessing techniques, n° %15, IEEE Sens. J. 15 (2014) 2479–2487.
[40] M.A. Razzaque, C. Bleakley, S. Dobson, Compression in wireless sensor networks: A survey and comparative evaluation, n° %11, ACM Trans. Sensor Networks (TOSN) 10 (2013) 5.
[41] D. Gurve, B. Saini, InduSaini, An Improved Lossless ECG Data Compression using ASCII Character Encoding, chez IEEE WiSPNET, 2016.
[42] H. Moustafa, E.M. Schooler, G. Shen, S. Kamath, Remote monitoring and medical devices control in eHealth, 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, USA, 2016.
[43] M. Al disi, H. Djelouat, A. Amira, F. Bensaali, The accuracy and efficacy of real time compressed ECG signal reconstruction on a heterogeneous multicore edge-device, 21st Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 2018.
[44] K. Kanoun, H. Mamaghanian, N. Khaled, A real-time compressed sensing-based personal electrocardiogram monitoring system, 2011 Design, Automation & Test in Europe, Grenoble, France, 2011.
[45] F. Pareschi, M. Mangia, D. Bortolotti, A. Bartolini, L. Benini, R. Rovatti, G. Setti, Energy analysis of decoders for rakeness-based compressed, n° %16, IEEE Trans. Biomed. Circuits Syst. 11 (2017) 1278–1289.
[46] M. Al disi, H. Djelouat, A. Amira, B. Faycal, ECG signal reconstruction on the IoT-gateway and efficacy of compressive sensing under real-time constraints, IEEE Access 6 (2018) 69130–69140.
[47] Realtime Technologies Ltd, «Shimmer User Manual,» 2017. [En ligne]. Available: http://www.shimmersensing.com/images/uploads/docs/Shimmer_User_Manual_rev3p.pdf. [Accès le 24 july 2019].
[48] ShimmerResearch, «shimmer3,» 27 August 2018. [En ligne]. Available: https://github.com/ShimmerResearch/shimmer3/tree/master/LogAndStream. [Accès le 24 July 2019].
[49] Realtime Technologies Ltd, «ECG User Guide,» 2017. [En ligne]. Available: http://www.shimmersensing.com/images/uploads/docs/ECG_User_Guide_Rev1.12.pdf. [Accès le 24 July 2019].
[50] J. Ivković, A. Veljović, B. Randjelovic, V. Veljović, ODROID-XU4 as a desktop PC and microcontroller development boards alternative, 6th International Conference Technics and Informatics in Education (TIO2016), Čačak, Serbia, 2016.
[51] Armadillo, C++ library for linear algebra & scientific computing, 2018. [En ligne]. Available: http://arma.sourceforge.net/. [Accès le 30 07 2019].

[52] G. ZeroMQ, iMatix Corporation, 2018. [En ligne]. Available: http://zeromq.org/. [Accès le 30 07 2019].
[53] Google, Protocol Buffers, Google, 2018. [En ligne]. Available: https://developers.google.com/protocol-buffers/. [Accès le 29 July 2019].
[54] J. Zhang, Z. Gu, L. Yu, Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted ‘ 1 minimization reconstruction, IEEE J. Biomed. Health Inf. 19 (2015) 520–528.
[55] Firebase, "Choose a Database: Cloud Firestore or Realtime Database, Google, 31 01 2019. [En ligne]. Available: https://firebase.google.com/docs/database/rtdb-vs-firestore?authuser=1. [Accès le 30 07 2019].
[56] B. Byfield, Configuring Bluetooth devices with bluetoothctl, Blue Control, 2017. [En ligne]. Available: http://www.linux-magazine.com/Issues/2017/197/Command-Line-bluetoothctl#article_i1. [Accès le 25 July 2019].
[57] H. Djelouat, M. Al Disi, A. Amira, F. Bensaali, X. Zhai, Compressive Sensing Based ECG Biometric System, Intelligent Systems and Applications, Cham, Springer International Publishing, 2019, pp. 126–137.
[58] H. Djelouat, X. Zhai, M. Al Disi, A. Amira, F. Bensaali, System-on-chip solution for patients biometric: a compressive sensing-based approach, IEEE Sens. J. 18 (2018) 9629–9639.