System-on-Chip Solution for Patients Biometric: A Compressive Sensing-Based Approach

Hamza Djelouat, Xiaojun Zhai, Mohamed Al Disi, Abbes Amira, *Senior Member, IEEE,*
and Faycal Bensaali, *Senior Member, IEEE,*

Abstract—The ever-increasing demand for biometric solutions for the internet of thing (IoT)-based connected health applications is mainly driven by the need to tackle fraud issues, along with the imperative to improve patient privacy, safety and personalized medical assistance. However, the advantages offered by the IoT platforms come with the burden of big data and its associated challenges in terms of computing complexity, bandwidth avail-ability and power consumption. This paper proposes a solution to tackle both privacy issues and big data transmission by incorporating the theory of compressive sensing (CS) and a simple, yet, efficient identification mechanism using the electro-cardiogram (ECG) signal as a biometric trait. Moreover, the paper presents the hardware implementation of the proposed solution on a system on chip (SoC) platform with an optimized architecture to further reduce hardware resource usage. First, we investigate the feasibility of compressing the ECG data while maintaining a high identification quality. The obtained results show a 98.88% identification rate using only a compression ratio of 30%. Furthermore, the proposed system has been implemented on a Zynq SoC using heterogeneous software/hardware solution, which is able to accelerate the software implementation by a factor of 7.73 with a power consumption of 2.318 W.

Index Terms—Internet of things (IoT), compressive sensing (CS), zynq SoC, reconstruction algorithms, pattern recognition.

I. INTRODUCTION

With the aging world population, the rising cost of health-care has put many challenges to governments and healthcare providers. Thus, strong urge to explore all the available com-munication infrastructures and powerful computing embedded platforms to support an unobtrusive, reliable and cost effective patient remote monitoring in order to provide sustainable, secure and personalized services to patients.

Wearable technology and continuous monitoring using the internet of things (IoT) is causing an explosion in biometric data. The latter can be combined with other factors such as medical history records to better understand the patient's con-dition and enhance the provided clinical assistance. However, IoT-based connected health applications face issues related to both big data transmission/processing and user privacy [1].

In a typical monitoring scenario, each patient uses a partic-ular sensing device which is linked to a unique identifier in order to be recognized at cloud server [2]. However, different users can use the same device resulting in merging the health record for different patients into one single labelled record which can lead to serious inaccuracies in the medical diagno-sis, hence, causing life threatening problems. In addition, the accuracy of these records is imperative for health organizations to keep track of individual patients and manage the overall patient population. Therefore, patients identification within the

continuous monitoring platforms will reduce misdiagnoses and increase effective treatment.

Thus, offering a solution that can address both big data issues and user privacy challenges is an imperative task to address for the next generation of remote monitoring plat-forms. This paper puts forth a platform for wireless moni-toring that offers an identification mechanism as well as it provides energy optimization by compressing the data before transmission. The proposed solution is based on two pillars, compressive sensing (CS), to reduce the number of samples transmitted and a simple, yet, effective pattern recognition approach to identify different individuals.

CS [3], [4] is an emerging paradigm that has gained con-siderable attention among signal processing community. CS aims to capture a compressed form of sparse signals directly without going through traditional sampling and compression paradigms. To this end, rather than collecting the values of uniformly spaced points in the signal, random compressive measurements are collected, where each measurement value correspond to the weighted summation of the entire signal. In addition, the compressed data capture enough information about the signal which enable a near-optimal recovery for the original data using sophisticated reconstruction algorithms, for instance, orthogonal matching pursuit (OMP) [5]. CS aims at shifting the complexity from the sensors which are usually resources constrained and self powered to the receiver side which is usually installed on computing platforms with relaxed constraints.

On the other hand, identification systems rely on pattern recognition mechanisms in order to match a given data signa-ture from a particular individual against a dataset of acquired signatures. Identification systems based on biometric traits rely on the unique and intrinsic behavioral (gait, interaction gestures and voice) and/ or physiological (DNA, Iris, face and electrocardiogram (ECG)) characteristics of individuals. Ide-ally, an identification system should grant an easy, universal, unique and permanent identification. However, an acceptable biometric system can satisfy relaxed constraints depending on the requirements and the purpose of the application.

ECG-based identification can be traced back to the work of Biel *et al.*[6] and irvine et al. [7]. The main idea is to use the ECG characteristics which differ from a person to another as a biometric measure to distinguish between different individuals. The key stage in ECG based identification is to extract the most significant information from the ECG by means of feature selection techniques. Feature selection aims at identifying the features that provides significant information

about the data and discard the features that are irrelevant and do not contain any discriminatory information.

To this end, two approaches could be distinguished: (i) characteristic-based methods [8] that use the ECG fiducial points and (ii) waveform-based methods [9] in which the morphology of entire ECG segment is analyzed. Characteristic-based features represent the fiducial points of the ECG which correspond to the P, QRS, T waves [10]. Fiducial points are points of interest within the heart electrical activities. The amplitude, duration and the correlation between these points can be used as reference for individual identification [6], [7], [11]. On the other hand, non-fiducial approaches are used to extract discriminatory information from the ECG segment either with partial detection of the fiducial points [9], [12], [13], [14], [15] or by using the entire ECG segment to extract the salient information from the data without detecting any of the R peaks [16].

In this work we present a unified framework for an ECG-based biometric system by exploiting the CS theory. The main objective is to present a SoC solution that reduces the overall power consumption while maintaining a high, reliable identification accuracy. The proposed framework is able to achieve 98.88% identification rate using only a compression ratio of 30% of the original signal size, as a result this would significantly reduce the energy transmission due to the smaller data size. In addition, as the proposed work is based on a heterogeneous re-configurable SoC, it can be integrated together with other ECG monitoring applications without any additional sensory equipment. The main contributions of this paper are the following:

- Propose and model an ECG-based identification system by exploring the CS concept at the acquisition step. The proposed system is quantified in terms of reconstruction quality, identification rate and execution time in order to confirm with real-time requirements.
- Introduce a way to integrate the CS and identification unit into heterogeneous re-configurable hardware. This allows the implementation of a highly adaptive and scalable high-performance data processing system.

The remaining sections of this paper are organized as follow, section II provides a brief overview of CS concept and its associated reconstruction algorithms. Section III presents the modelling and software implementation results for the proposed platform. In section IV, a detailed analysis of the OMP algorithm and its optimized architecture and its hardware implementation. Section V discusses the hardware implementation results. Section VI concludes the paper.

II. COMPRESSIVE SENSING OVERVIEW

The conventional acquisition paradigm samples the signal based on its bandwidth rate, a particular signal with a bandwidth of W should be sampled at least at frequency $f_s \geq 2W$ to fully preserve its information. Usually, the collected data are big and highly redundant, thus, the signal is often compressed by discarding the redundancy within the signal [17]. The compression would benefit the whole system by reducing the delays and the cost of the wireless communication as well as simplifying the computation process.

Subsequently, to avoid acquiring huge amount of redundant samples, a very interesting theory, namely CS, has been proposed in [3], [4]. CS suggests that it is possible to surpass the limitation of Nyquist-Shannon sampling paradigm by leveraging a particular structure that wide class of real world signals exhibit. CS states that if the information rate of the signal is less than its bandwidth rate, i.e., signal is either sparse or compressible, than it is possible to capture all the information without any loss using a fewer number of samples than what the Nyquist-Shannon theorem states. Hence, CS enables a potentially large reduction in the sampling and computation costs for sensing signals with sparse or compressible behavior [18].

A. CS Encoder

Consider the orthogonal basis $\{\Psi_i\}_{i=1}^n$ that spans $\mathbb{R}^n$, any signal $x \in \mathbb{R}^n$ can be expressed as a linear combination of the elements of Ψ with the elements of the vector $s = [s_1, s_2, \cdots, s_n]^T$ such that $x = \sum_{i=1}^n \Psi_i s_i$. The signal x is said to be k-sparse if the vector s has only $k \ll n$ non-zero entries. The matrix Ψ is called the sparsifying matrix in the context of CS. For instance, discrete cosine transform (DCT) and discrete wavelet transform (DWT) have been widely used as sparsifying basis for bio-medical signals [19].

The CS encoder performs both acquisition and compression simultaneously to acquire directly a compressed form of the signal. In the CS encoder, the input signal x is multiplied by a tall random sensing matrix, hence, the acquisition process in CS can be modeled by:

$$y = \Phi x = \Phi \Psi s = \Theta s \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$ represents the sensing matrix used to acquire and compress the data, Moreover, The ratio $\frac{m}{n}$ is defined as the compression ratio (CR).

The design of the sensing matrix should comply with two conditions, namely, incoherence and the restricted isometry property (RIP) [20]. Incoherence guarantees that each sparse vector $\mathbf{x}$ is mapped into a dense vector $\mathbf{y}$, i.e. each sensing sequence is able to collect part of the signal information content. Whereas, the RIP condition guarantees that mapping $\mathbf{x}$ from an n-dimensional space to m-dimensional space preserves the ℓ_2 norm of the signal and the distance between any two distinct signals $\mathbf{x}$ and $\mathbf{x}'$ is preserved. Both conditions can be guarantees by exploring random matrices [21]. A matrix Φ with entries drawn from an independent and identical distribution holds both the RIP and the incoherence conditions with any fixed sparsifying basis Ψ. In addition, Bernoulli matrices hold the same level of guaranties [22].

B. CS Decoder

Fast and efficient reconstruction algorithms are keys to incorporate CS in real world applications, thus, several reconstruction algorithm classes have been proposed in the literature. Nevertheless, the well recognized algorithms falls under the umbrella of two major algorithmic approaches, namely, convex optimization and greedy algorithms.

1) Convex Optimization: Convex optimization are based on ℓ_1-minimization and they were proposed as an alternative to the ℓ_0 minimization solver as it is an NP-hard problem for any large matrix $\boldsymbol{\Phi}$ [23]. The ℓ_1-minimization approach, known as basis pursuit (BP) [23], considers the following solution:

$$\hat{\boldsymbol{x}} = arg \min \|\boldsymbol{x}\|_1 \quad subject \quad to \quad \mathbf{y} = \boldsymbol{\Phi x} \qquad (2)$$

A noisy acquisition model adapts itself by considering either basis pursuit denoising (BPDN) solution [24] or least absolute shrinkage and selection operator (LASSO) approach [25] depending on the apriori knowledge of the noise level.

2) Greedy Algorithms: Greedy algorithms have been widely exploited in CS applications due to their relatively simple framework which provides a fast reconstruction with a low implementation cost. These methods enhance iteratively the approximation for the signal by making locally optimal choices. Greedy algorithms consist of two main steps, element(s) selection and coefficients update. These methods are usually initialized with a residual $\boldsymbol{r}^{[0]} = \boldsymbol{y}$, signal estimate $\hat{\boldsymbol{x}}^{[0]} = 0$ and empty support set $\varGamma = [\ ]$. At each iteration j, single or multiple elements from the sensing matrix are added to the support set, the signal estimate is calculated using a least square approach $\boldsymbol{x}^{[j]} = \boldsymbol{\Phi}_T^\dagger \boldsymbol{y}$. Additionally, the residual is minimized $\boldsymbol{r}^{[j]} = \boldsymbol{y} - \boldsymbol{\Phi}_T \boldsymbol{x}^{[j]}$. The algorithms halt when the residual norm is smaller than a predefined threshold.

Currently, the well established greedy algorithms include OMP [5], compressive sampling matching pursuit (CoSaMP) [26] and subspace pursuit (SP) [27].

III. PROPOSED SYSTEM AND ITS SOFTWARE IMPLEMENTATION

The main motivation of the proposed system is to tackle two issues facing patients monitoring, which are high power consumption and user privacy. Therefore, the key features provided by the proposed system are, 1) data transmission load reduction and 2) high-quality identification rate performance. To this end, two blocks are implemented, CS-based coding/decoding and identification block using machine learning algorithms. The necessity to compress ECG in remote monitoring applications is well established [28]. Furthermore, CS has been proven to outperform state-of-the-art compressing techniques in several setups in terms of power consumption, high compression and reconstruction quality [29], [30], [31], [32].

The CS-based coding stage consists of compressing each 2-sec ECG before transmission. Since the ECG signal is already acquired, we apt for digital CS [33]. This approach is performed by multiplying the signal by a tall sensing matrix $\varPhi \in \mathbb{R}^{M \times N}$. In the design of sensing matrix, the elements of $\varPhi$ have been drawn from three different distributions, namely, random, Bernoulli and sparse distribution. In the case where $\varPhi$ is a sparse matrix, the approaches can be seen as a non-uniform sampler (NUS) [34], where the CS encoder picks an M samples randomly from the whole N dimension ECG signal. Exploring the NUS architecture would reduce significantly the implementation complexity of the CS encoder [35]. On the other hand, decoding is a straightforward approach where

the original N-dimensional ECG is reconstructed from the M-dimensional signal using OMP algorithm. Furthermore, the usefulness of the reconstructed ECG data is evaluated in terms of one particular application, namely, ECG-based patient identification.

A. System Overview

The proposed CS-based identification is presented in Fig. 1. The system comprises of compression stage, a reconstruction phase and a pattern recognition unit. The latter consists of ECG pre-processing, features extraction and identification. Fig. 1. illustrates the overall framework of the investigated system.

The experimental data used to quantify the proposed system are obtained from PhysioNet-MIT Arrhythmia Database [36], which offers free web access to multi-channel ECG. The database contains recordings (30 minutes) sampled at 360 Hz with 11 bits precision. Each record contains normal and abnormal heart beat morphologies. The MIT Arrhythmia Database were collected from 47 subjects (Records 201 and 202 came from the same male subject). Fig. 2, presents three different recordings collected from the MIT-arrhythmia database.

Afterward, the ECG dataset has been divided into two subsets, the first one is used for extracting the features in order to train the identification model and the second one is used to validate and test the performance of the proposed platform. The training features set is extracted directly from the original ECG signal without undergoing any compression phase. Whereas, the testing ECG data are first compressed, transmitted to the CS decoder in order to be reconstructed using OMP algorithm. Afterward, the classification is performed and quantified in terms of true identification rate and real-time processing.

B. Reconstruction quality

In order to assess the reconstruction quality, the percent root-mean-square difference (PRD) has been used to quantify the amount of distortion in the reconstructed ECG signals. The PRD can be linked to the quality of reconstructed ECG for medical diagnosis through the work of Zigel *et al.* [37], where a PRD value less than 9% is considered acceptable for physician based diagnosis. However, a less restricted condition can be considered for computer based medical diagnosis applications such as heart rate estimations [38]. The PRD is calculated as follow:

$$PRD(\%) = 100 \times \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$$

Such that $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the original and the reconstructed ECG signals, respectively.

In addition, a major challenge in any biometric system is the real-time constraint. Thus, in order to achieve real-time performance, all the processing and computation to identify one ECG segment should be performed before the next ECG segment arrives to the the decoder. Therefore, in this work a real-time reconstruction window of 1 sec has been established.
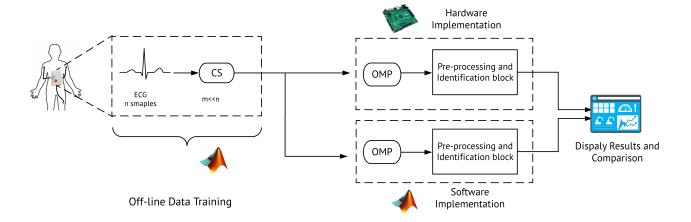
Fig. 1: Overall system diagram. The proposed system consists of an offline compression block and heterogeneous software/hardware reconstruction and identification blocks.
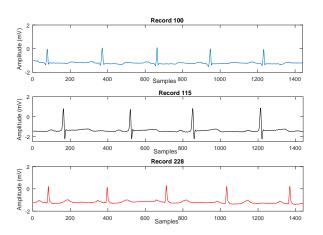

Fig. 2: Samples from recording 100, 115 and 228 [36]. The plots represents a 4-sec recordings from three different patients.

Subsequently, in order to present trustworthy results, MAT-LAB Monte Carlo simulations have been conducted over the whole data contained in the MIT-arrhythmia database. a 1-min recording has been collected from each subject and each subject recording has been divided into 60 ECG segments of 360 samples. An overall of 2880 reconstruction processes have been conducted. The orthonormal Symlet-4 wavelet matrix has been adopted as sparsifying basis. In addition, three different settings have been considered in the design of the sensing matrix by using a dense random matrix, a sparse matrix and a Bernoulli matrix.

Figure 3 illustrates the obtained performance in terms of PRD. Furthermore, it presents the averaged reconstruction time for the three different settings. The reconstructed time presented is required for an ECG window of 1 sec. It is worth mentioning that the operating system where the algorithms have been implemented is 64-bit Windows 7 professional, with a processor of an intel core i7-3770 @ 3.4Ghz CPU and a

RAM of 16.0 GB.

From Figure 3, the following observations can be drawn:

- Intuitively, by increasing the number of samples, the obtained results show an improvement in the reconstruction quality. In addition, a good reconstruction quality is achieved around a CR of 0.3 where the achieved PRD is about 7%. The obtained results show inferior performance compared to some recent studies for CS-based ECG reconstruction [28], [32], [39]. However, to establish a fair comparison, the same ECG data should be used. Therefore, besides the average PRD achieved using all the ECG signal in the MIT-arrhythmia database, we run several independent analysis to compare our obtained findings with results presented in the literature using the same ECG signal records. In [28], the author used Records 107, 118, and 119 and they have achieved a PRD of 2.6 for a CR=0.4. Subsequently, using the same records, our results revealed a PRD of 2.66, 2.7 and 2.55 for Bernoulli, random and sparse matrices, respectively, using the same values of CR. In [32], the authors used the record 107 only, the results obtained showed a very good recovery (PRD<2) using only CR=0.4. In our analysis, the best-obtained PRD=3.5 is achieved using a random matrix. Moreover, in [39], authors selected randomly a 100 segment form the entire MIT database, where each segment has a length of N=512. The results showed a PRD < 9 for a CR= 0.6. However, it is worth mentioning that the obtained results showed to have similar results with the only work in the literature that stated that the entire MIT-Arrhythmia has been used for compression/reconstruction [40]. For a CR=0.25, the proposed approach in [40] achieved a PRD=9 compared to PRD=10 achieved by our proposed approach. Table I provides a summary and a comparison between the proposed work and the best case performance presented in the literature in terms of PRD.
- The design of the sensing matrix does not have a remarkable impact on the reconstruction performance. However,

TABLE I: Summary of ECG reconstruction approaches performances in terms of PRD

Reference	Record Used	CR	PRD
[28]	107, 118, and 119	0.4	2.6
Proposed work	107, 118, and 119	0.4	2.55
[32]	107	0.4	< 2
Proposed work	107	0.4	3.5
[41]	100, 102, 230, and 109	0.4	< 2
Proposed work	100, 102, 230, and 109	0.4	2.5
[40]	All records	0.3	7
Proposed work	All records	0.3	7.5

a slight improvement on the reconstruction speed can be achieved using sparse sensing matrix in the high quality reconstruction regime, i.e, for CR values greater than 0.35. Therefore, it can be argued that using a Bernoulli and sparse matrix would be beneficial in the design of both CS encoder (the compression operation) [22] and the CS decoder (based on the obtained results).
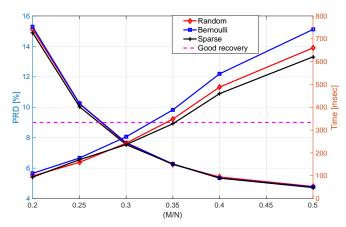

Fig. 3: Reconstruction quality for different sensing matrix setting. The reconstruction quality is quantified in terms of PRD (PRD < 9 indicates a good recovery) and recovery time.

In addition, we elaborate in particular the performance of the proposed CS-based compression scheme for the abnormal beats in the used ECG database in order to assess its ability to achieve similar results as stated in Figure 3. Thus, from the MIT database [36], we select the Records 106 and 201 for the testing. These records are observed to include a remarkable number of Premature ventricular contraction (PVC) beats. In fact, the Records 106 and 201 show to have 520 and 198 PVC beats respectively.

Table II shows the obtained PRD for the two records with a CR value ranging from 0.3 to 0.55. For the sake of simplicity, only the results obtained using Bernoulli matrices are presented as sparse and dense random matrices achieved similar results. The obtained results revealed a drop in the quality of reconstruction for the two records. For instance, for the Record 107 that has only 57 PVC heartbeats, a PRD < 2 is achieved for a CR of 0.4 compared to 6.87 and 5.34 for the records 106 and 201 respectively. Nevertheless, although even with this inferior quality of signal, CS can still be efficiently applied to achieve a good recovery quality using only a CR of 0.3.

TABLE II: Reconstruction performance for PVC heartbeats in terms of PRD

Record	Compression ratio					
	0.3	0.35	0.4	0.45	0.5	0.55
106	8.20	7.58	6.87	6.40	5.83	5.33
201	6.66	6.10	5.62	5.34	4.88	4.76

C. Identification

The CS-based ECG biometric system stores the ECG training data in a matrix $\mathbf{T}_R \in \mathbb{R}^{J_T \times L}$ where J represents the number of training segments and L denotes the number of samples in each training set. Where each row of $\mathbf{T}_R$ presents a single ECG training segment. However, prior to building up the training set $\mathbf{T}_R$, all the ECG recordings undergo a reprocessing stage in order to remove the baseline wander (BW) and power line interference [42]. A $1 - 40$ Hz band-pass filter has been applied to each ECG recording, followed by the Pan-Tompkins algorithm [43] to detect accurately the R peaks. In fact, correct and reliable personal identification based on ECG treats depends mainly on the significant discriminatory features that have to be extracted from the ECG signal. In the proposed system, the fiducial-based feature extraction approach has been used [44]. In such approach, the most important samples are the ones located within the PQRST wave. In a typical ECG, the duration of the PQRST wave spans about 0.4 sec [45] . Therefore, for an ECG sampled at 360 Hz, the PQRST wave should contain around 120-140 samples. Subsequently, in order to detect these samples, the R peak should be detected first and a window of L samples that includes the samples around it should be considered.

On the other hand, the compressed testing data are first reconstructed, then, the same approach for pre-processing and feature extraction are applied to build up the testing data-set $\mathbf{T}_M \in \mathbb{R}^{J_M \times L}$.

To further quantify the quality of the CS-based identification, the validity of the reconstructed data for individual identification has been carried out. To this end, the K-nearest neighbour algorithm (KNN) [46] has been explored. KNN represents one of the widely used classification techniques in machine learning problems. KNN requires two data sets as an input, the training data $\mathbf{T}_R$ set and the label data set $C_T \in \mathbb{R}^{J_T \times 1}$ which assign a unique class label for the ECG training segments from the same individual. Then, for each ECG testing segment, KNN compares the k distances between the given test segment and its k nearest training segments. Consequently, the test segment is assigned with the same class as the class of the majority of its k nearest neighbours. KNN algorithm depends on the parameter k, this coefficient determines how many neighbours influence the classification. If $k = 1$, the case is simply assigned to the class of its nearest neighbour.

Table III presents the successful identification rate accuracy achieved by using KNN algorithm as a classifier. To provide a consistent evaluation for the validity of the reconstructed data for classification, the testing data were used for identification with and without compression. For the latter, the achieved identification accuracy is presented in the column labelled original data. It is worth mentioning that for KNN,

the algorithm parameter is set to be $k = 1$. In addition, the length of features set segment has been set to three different value $L = \{51, 71, 101\}$. The obtained results shows the ECG biometric system achieves a high identification rate up to 98.88%.

Moreover, it is clear that increasing the number of samples in the acquired data from each individual would render a better reconstruction, and subsequently a high identification rate is achieved. However. for the low CR range i,e., $CR < 0.2$, the identification quality is unacceptable, this can be explained by the fact that the R peaks can not be detected properly due to the highly distorted reconstructed data. Moreover, by using a 35% of the samples a near optimal accuracy is achieved and a maximum of 3.2% , 5.4% and 3.7 % drop-down is obtained using only 25% of samples for $L = 101, L = 71$ and $L = 51$, respectively, compared to classification accuracy using directly the data without compression. In Overall, the achieved identification rate from the compressed data is similar to the existing methods used the original ECG signals, which have identification rates 98% [42], 99.6% [47] and 99.5% [44] respectively.

TABLE III: Identification rate for different testing periods

L	Original Data	Compression ratio					
		15	20	25	30	35	40
51	95.68	49.05	78.64	91.96	94.37	95.6	95.68
71	97.27	47	83.83	91.89	95.41	97.27	97.27
101	98.88	54.65	88.37	95.65	96.37	97.82	98.55

Furthermore, Table IV quantifies the performance of the system using different values of $k = \{1, 2, 3, 4\}$ using $L = 101$. The reported results show that shows that selecting only the single nearest training sample ($k = 1$) to assign the class of the testing segments provides the highest identification rate compared to the other cases.

TABLE IV: Identification rate using different values of k

k	Compression ratio			
	25	30	35	40
$k = 1$	95.65	96.37	97.82	98.55
$k = 2$	92.02	95.65	97.10	97.82
$k = 3$	89.85	94.92	97.82	98.55
$k = 4$	89.13	94.92	96.37	96.37

D. Execution time

Furthermore, enabling real-time processing is an imperative constraint to fulfill. Thus, for any continuous monitoring system, if the time required to process on ECG segment exceeds the real-time window, several issues would occur. Therefore, the proposed framework is quantified in term of the entire identification process for one single segment including reconstruction, pre-processing and classification. Figure 4 presents the normalized execution time over the real-time window. The results show that whatever, the CR value adopted, the entire identification process consumes at most 0.6 of the real time window. In addition, it is clear that the reconstruction time consumes the majority of the entire identification system. The reason is that the reconstruction process includes different blocks of matrix-vector multiplications and matrix inversion.

More details and analysis for the complexity and execution time for the different stages will be discussed in the following sections.
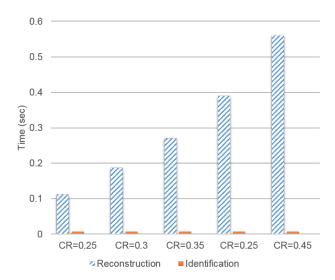

Fig. 4: Normalized execution time. The obtained results reveal that the reconstruction operation is the most time consuming process in the identification stage.

IV. HARDWARE IMPLEMENTATION

The proposed ECG recognition system consists mainly of the following four components: 1) ECG signal loading, 2) ECG signal reconstruction, 3) R peak detection, and 4) KNN classifier. The compressed ECG signals are firstly loaded, and then the OMP algorithm is applied on the compressed ECG signal in order to reconstructed the original ECG signals. Subsequently, the Pan-Tompkins is applied on the reconstructed signal to detect the R peaks. Finally, a KNN classifier is used to classify the data clusters. The proposed system has been modeled with C++ in Xilinx Vivado HLS environment [48]. In Vivado HLS, the proposed system is implemented in C++, and then synthesised and translated to a Hardware Description Language (HDL). A set of pragma directives have been used to optimize the codes for generating high performance hardware IP cores.

Since the signal reconstruction (OMP) block consumes the most of the processing time in the proposed system discussed in the previous section, an optimization scheme is proposed to implement the OMP algorithm on hardware to accelerate the process.

A. OMP analysis

The reconstruction is the process of solving equation (1) in order to find the sparse representation $\mathbf{s}$ and subsequently calculate the original ECG signal $\mathbf{x}$. OMP algorithm is initialized with a residual $\boldsymbol{r}^{[0]} = \boldsymbol{y}$, empty support set $\Gamma = \emptyset$ and maximum iteration number α. At each iteration i, the algorithm iteratively chooses one of the columns of $\boldsymbol{\Theta}$ that is mostly correlated with the residual $\mathbf{r}$ and then it removes the contribution of this column from $\boldsymbol{\Theta}$ to compute next residual.

A new estimate of the original signal is computed thereafter. After α iterations, the final estimate of the original signal is calculated.

Algorithm 1 : Orthogonal matching pursuit (OMP)

1. **Inputs**: The matrix $\boldsymbol{\Theta}$, measurements vector y, maximum iteration α.
2. **Outputs**: sparse signal approximation $\hat{s}$.
3. **Initialize**: $r^{(0)} = y$, $\Gamma^{(0)} = [\ ]$
4. **for** $i = 1, \cdots, \alpha$ **do**
5. $\Lambda^{(i)} = argmax_j |\langle \boldsymbol{\Theta}, r^{(i-1)} \rangle|$ {find k-column with highest correlation}
6. $\Gamma^{(i)} = \Gamma^{(i-1)} \cup \Lambda^{(i)}$ {concatenate k-column with Γ}
7. $\hat{s}^i = \boldsymbol{\Theta}^{\dagger}_{\Gamma^{(i)}} y$ {least squares optimisation}
8. $r^{(i)} = y - \boldsymbol{\Theta}\hat{s}^{(n)}$ {update the residual}
9. **end for**

The optimization problem (i.e. line 5 of Algorithm 1) is solved by calculating correlation vector **c** as follows:

$$\mathbf{c} = \boldsymbol{\Theta}^T r^{(i-1)} \tag{3}$$

where $r^{(i-1)}$ is the residual vector at $(i)^{th}$ iteration. The index of the component of c having the maximum absolute value is firstly located, and then the corresponding column is extracted from $\boldsymbol{\Theta}$ and added to the matrix $\Gamma^{(i)}$ of the extracted columns. After that, an error of the received and residual signals is calculated during each iteration. An estimate of the reconstructed signal **s** is finally obtained by solving the following equation:

$$\mathbf{y} = \boldsymbol{\Theta}_{\Gamma}\mathbf{s} \tag{4}$$

1) Complexity Analysis of OMP Algorithm: As it can be seen from the lines 5-8 of Algorithm 1, the OMP algorithm consists mainly of 5 functions: 1) finding the best fitting column (line 5); 2) update the existing matrix (line 6); 3) least square optimisation (line 7); 4) updating the residual (line 8); 5) solving equation (4). The computation of the first four functions is performed sequentially and repeated for each iteration, the fifth function is performed once the optimal reconstructed signal has been found, and this is meaning that the matrix inversion operation is only needed once, on contrast, the matrix multiplication is needed in every iteration. Therefore, in order to achieve the best overall performance and balance the hardware usage, we optimised the matrix multiplications from Function 1. In Function 1, the most computationally intensive task is the matrix multiplication in equation (3), where the size of the matrix $\boldsymbol{\Theta}$ is $m \times n$ and the length of residual vector **r** is m. The numbers of multiplications and additions are $m \times n$ and $m \times (n-1)$ respectively. In Function 2, the major operations are matrix assignments and comparisons, therefore, this is not a computationally intensive stage. In order to calculate the least square optimization in Function 3, a matrix multiplication is used, where the sizes of the matrices are $m \times n$ and $n \times m$ respectively. This requires $m^2 \times n$ multiplications and $m^2 \times (n-1)$ additions. Similar to Function 2, Function 4 mainly involves subtraction operations. Since Function 5 is performed once the optimal signal is found and is calculated only once, the most important parts of the overall complexity of OMP algorithm are resided in Function 1 and 3. Due to the iterative nature of the OMP

algorithm, it is difficult to allow parallel execution in the algorithm, and the functions inside iterations are data dependent. The computation of five functions cannot be performed concurrently. However, parallelism can be exploited at each function level by running the processes inside each function concurrently (e.g. matrix multiplications in Functions 1 and 3). Optimisations can be performed on all the operations to achieve full parallelism and the minimum latency. However, this will require huge resources and may not fit the size of the used hardware device. Therefore, a heterogeneous processing system consists of microprocessors and programmable logics has been chosen to implement the OMP algorithm, where the core computationally intensive tasks can be partitioned to hardware side for the implementation. The other tasks are then implemented on microprocessors. The details of the used hardware platform and implementation results are presented in the following section.

2) Optimisation of the OMP algorithm: As discussed in the previous section, equation (3) is the main operations in Function 1, let

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{0,0} & \theta_{0,1} & \theta_{0,2} & \cdots & \theta_{0,n-1} \\ \theta_{1,0} & \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,n-1} \\ \theta_{2,0} & \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_{m-1,0} & \theta_{m-1,1} & \theta_{m-1,2} & \cdots & \theta_{m-1,n-1} \end{bmatrix} \tag{5}$$

Hence, $\boldsymbol{\Theta}^T\mathbf{r}$ can be calculated as:

$$\boldsymbol{\Theta}^T\mathbf{r} = \left[\sum_{i=0}^{m-1} \theta_{i,0}\mathbf{r}_i, \sum_{i=0}^{m-1} \theta_{i,1}\mathbf{r}_i, \cdots \sum_{i=0}^{m-1} \theta_{i,n-1}\mathbf{r}_i, \right]^T \tag{6}$$

Pseudo code 1 is used to implement equation (6), where the loop pipelining is another key optimization technique that used to improve the throughput.

Pseudo code 1 :

1. **Inputs**: matrix $\boldsymbol{\Theta}$, $\mathbf{r}^{i-1}$ measurements is the residual vector at $(i-1)^{th}$ iteration.
2. **Outputs**: **c** is the correlation vector.
3. **for** $i = 1, \cdots, n$ **do**
4.   #pragma HLS PIPELINE
5.   # *pragma HLS ARRAY_PARTITION* variable=$\boldsymbol{\Theta}$ *block factor=f dim=1*
6.   # *pragma HLS ARRAY_PARTITION* variable=**r** *block factor=f dim=1*
7.   sum=0;
8.   **for** $j = 1, \cdots, m$ **do**
9. $sum = sum + \theta_{j,i} \times r_j$;
10.   **end for**
11. $c_i = sum$;
12. **end for**

One of the major limitation of block random access memory (BRAM) is that it has only two data ports. This significantly limits the throughput of a read/write (or load/store) from/to an array. The bandwidth could be improved by splitting the array (a single BRAM resource) into multiple smaller arrays (multiple BRAMs), that would effectively increase the number of ports. For example, the array $\boldsymbol{\Theta}_{m,j}$, can be partitioned into f small arrays in both dimensions by applying (*#pragma HLS ARRAY_PARTITION block factor=f*, where each array has size of $m/f \times n/f$. Similar strategy can be also applied on Function 3 as well. Figure 5 shows the partitioned arrays.
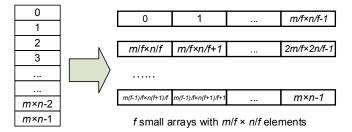
Fig. 5: Partitioned arrays. An array has been partitioned into f small arrays to increase the data throughput
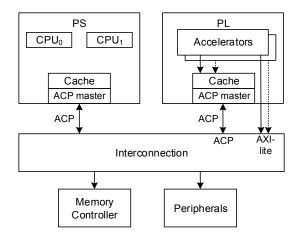

Fig. 6: An overview of the PL and PS system. The accelerator coherency port (ACP) has been used to interface the hardware co-processors and the rest of the system.

The proposed ECG recognition system has been implemented first in C++ and validated in Xilinx Vivado HLS environment. A set of pragma directives have been used to optimise the hardware implementation with the goal to achieve the optimal throughput with reasonable usage of hardware resources. In this paper, a Zynq-7000 SoC device is used for the proposed system. The proposed solution integrates the software programmability of an ARM-based processor with the hardware programmability of a FPGA. The IP cores are connected via advanced extensible interface 4 (AXI4) interfaces to the accelerator coherency port (ACP) of the ARM CPU as well as connecting through a Direct Memory Access (DMA) core in the programmable logic (PL) subsystem on the Zynq-7000 SoC device. In this work, SDSoC (v2017.2) has been used to interface the AP SoC PL hardware, the peripheral, the DMA engine, an AXI timer as well as other data mover logics [49]. The SDSoC is also used to design the AP SoC PS software to manage the peripherals and loading the testing data.

In Figure 6, the accelerator blocks have been implemented in Vivado SDSoC, which is used to generate efficient interfaces between the hardware co-processors and the rest of the system. Similar to the optimisation of using Vivado HLS, different interfacing options were explored through directive pragmas and code refactoring. Those IP blocks were integrated with other
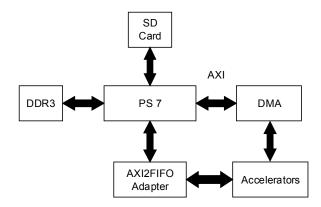

Fig. 7: Implementation overview. The accelerator is interconnected with the processing system (PS) via advanced extensible interface (AXI) interconnection blocks.

blocks of the design to be a heterogeneous embedded system. The hardware implementation of the accelerators used 32-bit floating-point arithmetic. As shown in Figure 7, the accelerator is interconnected with the PS via AXI interconnection blocks. The AXI2FIFO adapter block is used to convert the interface from AXI to FIFO in order to connect with the accelerators, and then the AXI DMA is used to move the processed data back to the PS. The implementation of the ECG recognition system was partitioned into two parts, one implemented on the PL and the other one of the PS. For example, the blocks of loading ECG signals, peak extraction and KNN classification shown in Figure 8 have been purely implemented on the PS7. On the other hand, the computationally intensive parts of the OMP algorithm have been implemented on the PL for acceleration. On the PS7 side, Linux operating system (OS) is used to support the I/O accesses of the application, and the open source Armadillo C++ linear algebra library is used to support matrix maths from the KNN classification [50]. Figure 9 summarizes the processing flow in terms of software and hardware data movements.

V. IMPLEMENTATION RESULTS

A. Vivado HLS Simulation

Prior to the hardware implementation, the proposed system was validated using Vivado HLS C++ simulator. Once the simulation has passed successfully, the C++ codes were translated to HDL, and then register transfer level (RTL) simulation is performed in order to validate the generated HDL architecture. The same C++ test bench used in the C/C++ simulation was used for the C/RTL co-simulation. A number of directive pragmas were used to guide the hardware generation in order to achieve the best throughput. Finally, the Vivado tool is used to complete the placement and routing of the proposed implementation. The hardware resource utilisation of the proposed implementation (i.e. Function 1 and 3) with different directive pragmas (i.e. Pseudo code 1) are shown in Figure 9.
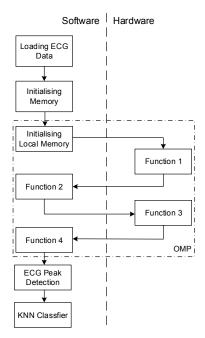
Fig. 8: An overview of the processing flow. The implementation of the ECG recognition system was partitioned into two parts, Function 1 and 3 have been implemented on the programmable logic (PL) and others are on the processing system (PS).

B. Implementation Results

The proposed system has been implemented on the Xilinx Zedboard, which is equipped with a Zynq-7000 All Programmable SoC XC7Z020-CLG484, 512 MB DDR3 memory and 16 GB SD card. In addition, the corresponding software (i.e. drivers, control codes and streaming phase) and hardware are partitioned and implemented using the Xilinx SDSoC development environment. In the proposed implementation,

As it can be seen from Figure 9, since all the arrays in the loops have been partitioned with different factors, the usage of hardware resources is dominant. In this work, the size of the input matrices are $m \times n$ and $n \times m$, where $m = 102$ and $n = 256$. The usage of flip-flops (FFs) and lookup tables (LUT) has increased significantly when the factor of partition is greater than 8. The usage of BRAM is almost stable when the factor is greater than 2. However, the use of the DSP48E units has gradually increased and the latency of the IP block is dramatically decreased when the factor is increased. However, the processing speed of this implementation is limited by the array partition factor and pipeline pragmas, it improves the latency by increasing the partition factor and as expected using much more hardware resources. Although the factors of array partitions have been increased accordingly and more hardware resources have been used, however, the processing speed has not improved significantly and the required hardware resources for both implementations are significantly higher than others. Therefore, the partition factor = 8 has been chosen for the final hardware implementation in order to overcome the latency and hardware utilisation trade-off.

TABLE V: Hardware resources usage of the proposed implementation of functions 1 and 3

Resource	Used	Total	Utilisation (%)
DSP48E	95	220	43
BRAM	112	140	80
LUT	53190	53200	99
FF	68257	106400	64

TABLE VI: Power consumption estimation of the proposed implementation

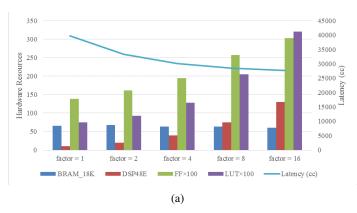| Utilization Details		Power (W)	Utilization (%)
Dynamic Power Consumption	Clock	0.120	5%
	Signals	0.189	8%
	Logic	0.092	4%
	BRAM	0.242	10%
	DSP	0.034	1%
	MMCM	0.106	5%
	PS7	1.535	67%
Static Power Consumption	Device Static	0.18	7%

the lengths of ECG signal and residual vector is 256 and 102 respectively. The hardware resources usage for the two functions (i.e. 1 and 3) is shown in Table V.

The hardware utilization results are based on the implementation using the pipeline and array partition pragma on a Zynq-7000 xc7z020 SoC. Based on the utilisation report, 99 (%) of the LUTs, 43(%) of the DSP48E and 64 (%) of the FFs are used to implement the matrix multiplications within the functions 1 and 3 (i.e. Pseudo code 1), these functions contain which have been implemented in parallel, it uses most of the resources. It is worth noting that the target Zynq SoC has a relatively small chip capacity in Xilinx 7 series family, which means that the proposed architecture is area-efficient, and can be easily deployed on a low-cost FPGA or integrated on a large chip.

1) Power Consumption: The details of the estimated power consumption of the implementation are summarised in Table VI. The PS7 consumes more power than the PL; this is due to the fact that the ARM dual core Cortex-A9 based processing system has much higher running frequency than the PL and it runs other parts of the application, drivers and control programs. Compare to the PS7, the custom logic blocks consume only a small portion of the total on-chip power consumption. In terms of power consumption, we obtained the power analysis from Xilinx Vivado [48], the total on-chip power consumption estimation for the implementation is 2.318 W, which includes 1.535 W and 0.783 W from the PS and the PL respectively.

TABLE VII: Processing time of the proposed implementation

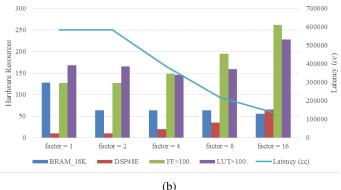|	Function 1 (Clock Cycles)	Function 3 (Clock Cycles)	Overall (Clock Cycles)
Hardware Accelerated Implementation	293100	1297913	718788320
Software Only Implementation	2260106	47703939	5558058708
Estimated Speedup	36.7	7.71	7.73

Fig. 9: The recourse utilizations of the proposed implementation. (a) Function 1: finding the best fitting column (b) Function 3: least square optimisation.

2) Timing Analysis: The running frequencies of the ARM processors and the PL are 650 and 148 MHz respectively. The processing time of the proposed system is measured by counting the number of clock cycles of the ARM processor that is used to wait for the results from the accelerators. Table VII shows the comparison between the software and hardware implementations of the accelerators in terms of the processing time in CPU clock cycles and millisecond. Based on the achieved results, the processing speeds of functions 1 and 3 are $0.45\ ms$ and $1.99\ ms$ respectively, which are 36.7 and 7.71 times higher than pure software implementation of the same functions. In addition, the overall processing time is 7.73 times faster than the software implementation. The proposed heterogeneous implementation uses 2.318 W, where only 0.783 W are consumed by the PL, and it is only 33% overhead than the pure software implementation.

VI. CONCLUSION

The combination of ECG biometric with CS concept presents an appealing technique that would grant reliable solution to both big data transmission and privacy issues facing the connected health applications.

The paper proposes a framework for patient identification that leverages a CS acquisition scheme and a simple and efficient machine learning tool in order to identify patient from their collected data, hence, provide patient-specific diagnosis and treatment.

The proposed approach introduces a way to integrate the CS decoder and identification unit into a heterogeneous reconfigurable hardware. This allows the implementation of a highly adaptive and high-performance data processing system. The obtained performance shows promising results, where a 98.88% identification rate is achieved using only compression ratio of 30%, which means that it only needs 30% of the original signal to achieve a significant identification accuracy. As a result of this, it would significantly reduce the energy transmission due to the smaller data size. Furthermore, the heterogeneous software/hardware implementation on the Zynq SoC platform is able to fasten the overall processing time by a factor of 7.73 with a cost of 2.318 W power consumption, which offers much better performance per watt compared with the pure software solution.

REFERENCES

[1] H. Baali, H. Djelouat, A. Amira, and F. Bensaali, "Empowering technology enabled care using iot and smart devices: A review," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 1790–1809, March 2018.

[2] P. Pierleoni, A. Belli, L. Maurizi, L. Palma, L. Pernini, M. Paniccia, and S. Valenti, "A wearable fall detector for elderly people based on ahrs and barometric sensor," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6733–6744, Sept 2016.

[3] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[6] L. Biel, O. Pettersson, L. Philipson, and P. Wide, "Ecg analysis: a new approach in human identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808–812, 2001.

[7] J. Irvine, B. Wiederhold, L. Gavshon, S. Israel, S. McGehee, R. Meyer, and M. Wiederhold, "Heart rate variability: a new biometric for human identification," in *Proceedings of the International Conference on Artificial Intelligence (IC-AI01)*, 2001, pp. 1106–1111.

[8] G. Wübbeler, M. Stavridis, D. Kreiseler, R.-D. Bousseljot, and C. Elster, "Verification of humans using the electrocardiogram," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1172–1175, 2007.

[9] K. N. Plataniotis, D. Hatzinakos, and J. K. Lee, "Ecg biometric recognition without fiducial detection," in *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*. IEEE, 2006, pp. 1–6.

[10] S. I. Safie, J. J. Soraghan, and L. Petropoulakis, "Electrocardiogram (ecg) biometric authentication using pulse active ratio (par)," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1315–1322, 2011.

[11] Y. Gahi, M. Lamrani, A. Zoglat, M. Guennoun, B. Kapralos, and K. El-Khatib, "Biometric identification system based on electrocardiogram data," in *New Technologies, Mobility and Security, 2008. NTMS'08*. IEEE, 2008, pp. 1–5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2018.2871411, IEEE Sensors Journal

11

[12] N. Venkatesh and S. Jayaraman, "Human electrocardiogram for biometrics using dtw and flda," in *20th international conference on Pattern recognition (icpr)*. IEEE, 2010, pp. 3838–3841.

[13] I. Odinaka, P.-H. Lai, A. D. Kaplan, J. A. O'Sullivan, E. J. Sirevaag, S. D. Kristjansson, A. K. Sheffield, and J. W. Rohrbaugh, "Ecg biometrics: A robust short-time frequency analysis," in *IEEE international workshop on Information forensics and security (wifs)*. IEEE, 2010, pp. 1–6.

[14] O. Boumbarov, Y. Velchev, and S. Sokolov, "Ecg personal identification in subspaces using radial basis neural networks," in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IEEE, 2009, pp. 446–451.

[15] J. Wao and Y. Wan, "Improving computing efficiency of a wavelet method using ecg as a biometric modality," *International Journal of Computer and Network Security*, vol. 2, no. 1, pp. 15–20, 2010.

[16] S. Gutta and Q. Cheng, "Joint feature extraction and classifier design for ecg-based biometric recognition," *IEEE journal of biomedical and health informatics*, vol. 20, no. 2, pp. 460–468, 2016.

[17] Y. Liu, M. De Vos, I. Gligorijevic, V. Matic, Y. Li, and S. Van Huffel, "Multi-structural signal recovery for biomedical compressive sensing," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2794–2805, 2013.

[18] S. A. Imtiaz, A. J. Casson, and E. Rodriguez-Villegas, "Compression in wearable sensor nodes: impacts of node topology," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1080–1090, 2014.

[19] D. Craven, B. McGinley, L. Kilmartin, M. Glavin, and E. Jones, "Compressed sensing for bioelectric signals: a review," *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 529–540, 2015.

[20] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, 2008.

[21] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013, vol. 1, no. 3.

[22] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing of eeg for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221–224, 2013.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[24] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[25] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[26] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[27] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2230–2249, 2009.

[28] A. J. Casson and E. Rodriguez-Villegas, "Signal agnostic compressive sensing for body area networks: Comparison of signal reconstructions," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 4497–4500.

[29] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 1, p. 5, 2013.

[30] F. Abate, V. K. L. Huang, G. Monte, V. Paciello, and A. Pietrosanto, "A comparison between sensor signal preprocessing techniques," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2479–2487, 2015.

[31] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1999–2008, 2013.

[32] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[33] D. E. Bellasi and L. Benini, "Energy-efficiency analysis of analog and digital compressive sensing in wireless sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 11, pp. 2718–2729, 2015.

[34] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[35] B. Bah and J. Tanner, "Vanishingly sparse matrices and expander graphs, with application to compressed sensing," *IEEE transactions on information theory*, vol. 59, no. 11, pp. 7491–7508, 2013.

[36] G. B. Moody and R. G. Mark, "The mit-bih arrhythmia database on cd-rom and software for use with it," in *Computers in Cardiology 1990, Proceedings*. IEEE, 1990, pp. 185–188.

[37] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (wdd) measure for ecg signal compression," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 11, pp. 1422–1430, 2000.

[38] D. Bortolotti, M. Mangia, A. Bartolini, R. Rovatti, G. Setti, and L. Benini, "An ultra-low power dual-mode ecg monitor for healthcare and wellness," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*. IEEE, 2015, pp. 1611–1616.

[39] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Structured sparsity models for compressively sensed electrocardiogram signals: A comparative study," in *Biomedical Circuits and Systems Conference (BioCAS), 2011 IEEE*. IEEE, 2011, pp. 125–128.

[40] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, "Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted ℓ_1 minimization reconstruction," *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 520–528, 2015.

[41] L. F. Polania and K. E. Barner, "Multi-scale dictionary learning for compressive sensing ecg," in *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE*. IEEE, 2013, pp. 36–41.

[42] I. Odinaka, P.-H. Lai, A. D. Kaplan, J. A. O'Sullivan, E. J. Sirevaag, and J. W. Rohrbaugh, "Ecg biometric recognition: A comparative analysis," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1812–1824, 2012.

[43] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE transactions on biomedical engineering*, no. 3, pp. 230–236, 1985.

[44] X. Zhai, A. A. S. Ali, A. Amira, and F. Bensaali, "Ecg encryption and identification based security solution on the zynq soc for connected health systems," *Journal of Parallel and Distributed Computing*, vol. 106, pp. 143–152, 2017.

[45] Y. N. Singh and P. Gupta, "Ecg to individual identification," in *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems(BTAS 2008)*. IEEE, 2008, pp. 1–8.

[46] A. Ait Si Ali, A. Amira, F. Bensaali, M. Benammar, M. Hassan, and A. Bermak, "Hw/sw co-design based implementation of gas discrimination," in *2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, July 2016, pp. 237–238.

[47] C. Ye, M. T. Coimbra, and B. V. Kumar, "Investigation of human identification using two-lead electrocardiogram (ecg) signals," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, 2010, pp. 1–8.

[48] "Xilinx inc. (2017). vivado hls user guide. available," www.xilinx.com.

[49] "Xilinx inc. (2017). sdsoc design guide. available:," www.xilinx.com.

[50] C. Sanderson, "Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments," 2010.