A Compressive Sensing Codec Architecture for ECG Signals with Adaptive Quantization and Stream Entropy Coding

SUBMISSION DATE / POSTED DATE

06-12-2022 / 11-12-2022

CITATION

Kumar, Shailesh; Prasad, Surendra; Lall, Brejesh (2022): A Compressive Sensing Codec Architecture for ECG Signals with Adaptive Quantization and Stream Entropy Coding. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.21677702.v1

DOI

10.36227/techrxiv.21677702.v1

A Compressive Sensing Codec Architecture for ECG Signals with Adaptive Quantization and Stream Entropy Coding

Shailesh Kumar, Surendra Prasad, *Senior Member, IEEE,* and Brejesh Lall, *Member, IEEE*

***Abstract*—Compressive sensing (CS) is quite appealing as a low-complexity method for the compression of ECG data in resource-limited wearable devices. This paper proposes a codec architecture comprising adaptive quantization and asymmetric numeral systems (ANS) based entropy coding of compressive measurements that can boost the compression ratio without sacrificing reconstruction performance. The quantized Gaussian entropy model for the compressive measurements is estimated directly from the data and is adapted dynamically to achieve better compression. We have tested our encoder with Block Sparse Bayesian learning as well as CS-NET sparse recovery algorithms on the MIT-BIH Arrhythmia database. Our encoder can achieve 5-25% of additional space savings over compressive sensing.**

I. Introduction

In wireless body area networks (WBAN) based telemonitoring networks [1], the energy consumption on sensor nodes is a primary design constraint [2]. The wearable sensor nodes are often battery-operated. It is necessary to reduce energy consumption as much as possible. It is desirable that a low-complexity encoder be used for the compression of ECG data from wearable devices. Compressive sensing (CS) [3]–[7] provides a very good solution to implement low-complexity encoders and has been extensively studied for ECG data compression [8], [9]. It uses a sub-Nyquist sampling method by acquiring a small number of incoherent measurements which are adequate to reconstruct the signal if the signal is sufficiently sparse in some basis. For a sparse signal $\mathbf{x} \in \mathbb{R}^n$, one would make m linear measurements where $m \ll n$ which can be mathematically represented by a sensing operation

$$\mathbf{y} = \Phi \mathbf{x} \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$ is a matrix representation of the sensing process and $\mathbf{y} \in \mathbb{R}^m$ the set of m measurements collected for $\mathbf{x}$. A suitable reconstruction algorithm can recover $\mathbf{x}$ from $\mathbf{y}$.

Ideally, the sensing process should be implemented at the hardware level in the analog-to-digital conversion (ADC) process. However, much of the use of CS in ECG follows a digital CS paradigm [10] where the ECG samples are acquired first by the ADC circuit on the device and then they are translated into incoherent measurements via the multiplication of a digital sensing matrix. These measurements are then transmitted to remote telemonitoring servers. A suitable reconstruction algorithm is used on the server to recover the original ECG signal from the compressive measurements. Reconstruction algorithms for ECG signals include: greedy algorithms [11] (simultaneous orthogonal matching pursuit), optimization-based algorithms [12], [10] (SPG-L1), Bayesian learning algorithms [13]–[15], and deep learning based algorithms [16].

We consider the problem of efficient transmission of compressive measurements of ECG signals over the wireless body area networks under the digital compressive sensing paradigm. Let $\mathbf{x}$ be an ECG signal and $\mathbf{y}$ be the corresponding stream of compressive measurements. Our goal is to transform $\mathbf{y}$ into a bitstream $\mathbf{s}$ with as few bits as possible without losing the signal reconstruction quality. A primary constraint in our design is that the encoder should avoid any floating point arithmetic.

A. Related Work

The literature on the use of CS for ECG compression is mostly focused on the design of a specific *sensing matrix*, *sparsifying dictionary*, or *reconstruction algorithm* for the high-quality reconstruction of the ECG signal from the compressive measurements. To the best of our knowledge, (digital) quantization and entropy coding of the compressive measurements of ECG data haven't received much attention in the literature.

Mamaghanian et al. [10] use a Huffman codebook which is deployed inside the sensor device. They don't employ any quantization of the measurements. The codebook is fixed. It cannot adapt to differing signal statistics. Simulation-based studies generally send the floating point compressive measurements to their decoder modules. They don't consider the issue of the number of bits required to encode each measurement. The compression ratio is often defined simply as $\frac{m}{n}$ or some variant of it [1]. The underlying assumption is that the measurements are encoded using the same number of bits as the original digital samples. Huffman codes have frequently been used in ECG data compression for non-CS methods [17], [18] for entropy coding. However, entropy coding of CS measurements has largely been ignored.

Asymmetric Numeral Systems (ANS) [19] based entropy coding schemes have seen much success in recent years for

Shailesh Kumar, Surendra Prasad and Brejesh Lall are with Department of Electrical Engineering and Bharti School of Telecommunication Technology And Management, Indian Institute of Technology, Delhi, India. Surendra Prasad is Honorary Professor and INSA Senior Scientist at IIT Delhi. e-mail: (bsz188670@dbst.iitd.ac.in)

[1] Other variants include $\frac{n}{m}$ [13], $\frac{n-m}{n} \times 100$ [16]

lossless data compression. They provide superior compression compared to Huffman like symbol codes. To the best of our knowledge, ANS stream codes have not been considered in the past for the entropy coding of compressive measurements.

B. Contributions

In this paper, we present an adaptive quantization and entropy coding scheme with a norm-bounded quantization noise for digital compressive measurements. A quantized Gaussian probability model [2] of the measurements is estimated directly from the data and the parameters for this model are used for the entropy coding of the measurements. Asymmetric numeral systems (ANS) based entropy coding is then used for efficient entropy coding of the quantized and clipped measurements. Our encoding scheme doesn't require a fixed codebook for entropy coding. The encoder can be implemented entirely using integer arithmetic.

C. Paper Organization

The rest of the paper is organized as follows. Section II describes our proposed codec architecture. Section III describes the ECG database used for the evaluation of the codec and the performance metrics. The analysis of the experimental results is covered in section IV. Section V summarizes our contributions and major findings.

II. Proposed Codec Architecture

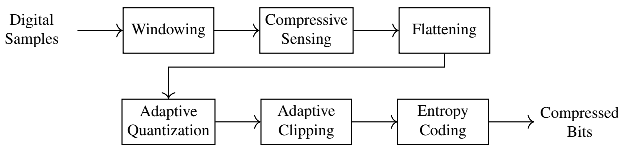


Fig. 1. Digital Compressive Sensing Encoder

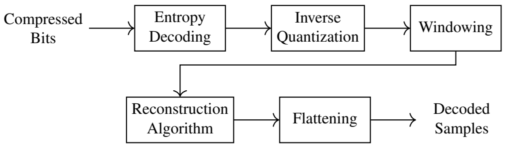


Fig. 2. Digital Compressive Sensing Decoder

This section describes a codec architecture for digital signals involving digital compressive sensing, quantization and entropy coding steps. Figure 1 and fig. 2 depict high-level block diagrams of the encoder and the decoder. The encoding algorithm is presented in algorithm 1. The decoding algorithm is presented in algorithm 2.

The digital signal is split into windows of n samples each. The windows of the digital signal are further grouped into frames of w windows each. The last frame may have less than w windows. The encoder compresses the digital signal frame by frame into a bitstream. It first sends encoding parameters in the form of a stream header (see table I). Then for each frame of the digital signal, it sends a frame header (see table II) followed by a frame payload consisting of the quantized and entropy-coded measurements for the frame. The decoder initializes itself by reading the stream header from the incoming bitstream. Then, it reconstructs the digital signal frame by frame.

Algorithm 1: Encoder algorithm

```
Send stream header ;
Build sensing matrix $\Phi$;
foreach frame of digital signal as $\mathbf{x}$ with $n_w$ windows do
    $\mathbf{X} \leftarrow$ window($\mathbf{x}$) ;
    // Sense
    $\mathbf{Y} \leftarrow \Phi\mathbf{X}$;
    $\mathbf{y} \leftarrow$ flatten($\mathbf{Y}$);
    // Adaptive quantization
    for $q = q_{\max} \dots q_{\min}$ (descending) do
        $\bar{\mathbf{y}} \leftarrow \lfloor \frac{1}{2^q}\mathbf{y} \rfloor$;
        $\tilde{\mathbf{y}} \leftarrow 2^q\bar{\mathbf{y}}$;
        if N_RMSE($\mathbf{y}, \tilde{\mathbf{y}}$) $\leq \rho$ then
            break ;
        end
    end
    // Quantized Gaussian model parameters
    $\mu_y \leftarrow \lceil$mean($\bar{\mathbf{y}}$)$\rceil$ ;
    $\sigma_y \leftarrow \lceil$std($\bar{\mathbf{y}}$)$\rceil$ ;
    // Adaptive range adjustment
    for $r = 2 \dots 8$ do
        $y_{\min} \leftarrow \mu_y - r\sigma_y$ ;
        $y_{\max} \leftarrow \mu_y + r\sigma_y$ ;
        $\hat{\mathbf{y}} \leftarrow$ clip($\bar{\mathbf{y}}, y_{\min}, y_{\max}$);
        if N_RMSE($\bar{\mathbf{y}}, \hat{\mathbf{y}}$) $\leq \gamma$ then
            break ;
        end
    end
    $\mathbf{c} \leftarrow$ ans_code($\hat{\mathbf{y}}, \mu_y, \sigma_y, y_{\min}, y_{\max}$) ;
    $n_c \leftarrow$ number of words in $\mathbf{c}$;
    Send frame header($\mu_y, \sigma_y, q, r, n_w, n_c$);
    Send frame payload($\mathbf{c}$);
end
```

Algorithm 2: Decoder algorithm

```
Read stream header ;
Build sensing matrix $\Phi$;
while there is more data do
    $\mu_y, \sigma_y, q, r, n_w, n_c \leftarrow$ read frame header ;
    $\mathbf{c} \leftarrow$ read frame payload ($n_c$);
    // Entropy model parameters
    $y_{\min} \leftarrow \mu_y - r\sigma_y$ ;
    $y_{\max} \leftarrow \mu_y + r\sigma_y$ ;
    $\hat{\mathbf{y}} \leftarrow$ ans_decode($\mathbf{c}, \mu_y, \sigma_y, y_{\min}, y_{\max}$) ;
    // Inverse quantization
    $\tilde{\mathbf{y}} \leftarrow 2^q\hat{\mathbf{y}}$;
    $\tilde{\mathbf{Y}} \leftarrow$ window($\tilde{\mathbf{y}}$) ;
    $\tilde{\mathbf{X}} \leftarrow$ reconstruct($\tilde{\mathbf{Y}}$) ;
    $\tilde{\mathbf{x}} \leftarrow$ flatten($\tilde{\mathbf{X}}$) ;
end
```

A. Sensing Matrix

Following [10], we construct a sparse binary sensing matrix Φ of size $m \times n$. Each column of Φ consists of exactly d ones and $m-d$ zeros, where the position of ones has been randomly

[2] A quantized Gaussian distribution is a quantized version of Gaussian distribution over bins of size 1 centered at integer values. It is defined by taking a Gaussian distribution with the specified mean and standard deviation, clipping it to an interval, renormalizing it to account for the clipped-off tails and then integrating the probability density over the bins [20].

TABLE I
STREAM HEADER

Parameter	Description	Bits
key	Pseudorandom generator key for Φ	64
n	Window size	12
m	Number of measurements per window	12
d	Number of ones per column in sensing matrix	6
w	Number of windows per frame	8
adaptive	Adaptive or fixed quantization flag	1
if (adaptive)		
ρ	N_RMSE limit for adaptive quantization	8
else		
q	Fixed quantization parameter	4
γ	N_RMSE limit for clipping	8

TABLE II
FRAME HEADER

Parameter	Description	Bits
μ_y	Mean value	16
σ_y	Standard deviation	16
q	Frame quantization parameter	3
r	Frame range parameter	4
n_w	Windows in frame	8
n_c	Words of entropy coded data	16

selected in each column. Sparse binary sensing matrices can be treated as adjacency matrix representations of bipartite graphs. In the equation $\mathbf{y} = \Phi \mathbf{x}$, let $\mathbf{x}$ and $\mathbf{y}$ represent node sets U and V respectively where each node in the node-set corresponds to a component of the vector. Consider the bipartite graph $G(\Phi) = (U, V, E)$ [3] such that an edge (i, j) belongs to E if and only if $\Phi(j, i) = 1$. Informally, such a graph is an expander [21] if each small subset of nodes in U has many neighbors in V. We refer to U as the left part and V as the right part of the graph G. A bipartite graph is called left d-regular if every node in the left part has exactly d neighbors in the right part. This corresponds to having exactly d ones in each column of Φ. A bipartite left-d regular graph $G = (U, V, E)$ is called a (k, d, ϵ)-expander if any set $S \subset U$ of at most k left nodes has at least $(1 - \epsilon)d|S|$ neighbors.

Sparse binary sensing matrices cannot satisfy the standard RIP property [22], however, they do satisfy a generalized RIP-p property [21], [23]. An $m \times n$ matrix Φ is said to satisfy RIP (p, k, δ) if for any k-sparse vector $\mathbf{x}$, we have

$$\|\mathbf{x}\|_p(1 - \delta) \leq \|\Phi \mathbf{x}\|_p \leq \|\mathbf{x}\|_p. \tag{2}$$

The adjacency matrices of expander graphs have this property. Gilbert et al. show in [21] that if Φ is the adjacency matrix of a (k, d, ϵ) expander graph $G = (U, V, E)$, then the scaled matrix Φ/d satisfies RIP $(1, k, \delta)$ property for $\delta = 2\epsilon$. They further show that for an expander matrix Φ, for any signal $\mathbf{x}$, given $\mathbf{y} = \Phi \mathbf{x}$, we can recover $\hat{\mathbf{x}}$ such that

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_1 \leq c(\epsilon) \|\mathbf{x} - \mathbf{x}_{|k}\|_1$$

where $\mathbf{x}_{|k}$ is the best k-term approximation of $\mathbf{x}$ and $c(\epsilon)$ is a constant depending on ϵ. Stable recovery guarantees are also available. Berinde et al. show in [24] that for k-sparse $\mathbf{x}$ and the measurements $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$ where the measurement noise $\mathbf{e}$ has an ℓ_1 bound, there exists an algorithm that recovers a k-sparse $\hat{\mathbf{x}}$ such that $\|\mathbf{x} - \hat{\mathbf{x}}\|_1 = \mathcal{O}(\|\mathbf{e}\|_1 / d)$ if Φ is a matrix induced by a (s, d, ϵ)-expander G where $s = \mathcal{O}(k)$. These theoretical results are the foundation of our encoder design wherein a bounded quantization noise doesn't impact the reconstruction quality.

Mamaghanian et al. [10] studied the variation of reconstruction SNR with d for SPGL1 algorithm [25]. They chose a value of $d = 12$ below which SPGL1 recovery suffered. Zhang et al. [13] showed experimentally that the BSBL-BO algorithm can do good recovery for much lower values of d. In our experiments, we have reported results for both $d = 12$ as well as $d = 4$.

B. Encoding

Here we describe the encoding process for each frame. Let a frame of digital signal be denoted by a vector $\mathbf{x}$. The frame is split into non-overlapping windows of n samples each ($\{\mathbf{x}_i\}_{1 \leq i \leq w}$). We put them together to form the (signed) signal matrix [4]:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_w \end{bmatrix}. \tag{3}$$

We perform compressive sensing on the whole frame of windows together as:

$$\mathbf{Y} = \Phi \mathbf{X}. \tag{4}$$

Note that by design, the sensing operation can be implemented using just lookup and integer addition. The ones in each row of Φ identify the samples within the window to be picked up and summed. Consequently, $\mathbf{Y}$ consists of only signed integer values.

Beyond this point, the window structure of the signal is not relevant for quantization and entropy coding. Hence, we flatten it into a vector $\mathbf{y}$ of mw measurements.

$$\mathbf{y} = \text{flatten}(\mathbf{Y}). \tag{5}$$

1) Quantization: The quantization for each frame is specified by a parameter q. This parameter is either fixed for the whole stream (as specified in the stream header), or varies from frame to frame (under adaptive quantization). It is given by:

$$\bar{\mathbf{y}} = \left\lfloor \frac{1}{2^q} \mathbf{y} \right\rfloor. \tag{6}$$

For integer measurement values, quantized values are also integers with a smaller range (by a factor of 2^q). It can be easily implemented on a computer as a signed right shift by q bits. We can measure the quantization error introduced by this step by comparing $\mathbf{y}$ with the inverse quantized values $\tilde{\mathbf{y}} = 2^q \bar{\mathbf{y}}$.

[3] U and V are left and right nodes of G and E is the set of edges between the left and right nodes.

[4] PhysioNet provides the baseline values for each channel in their ECG records. Since the digital samples are unsigned, we have subtracted them by the baseline value (1024 for 11-bit encoding). 11 bits mean that unsigned values range from 0 to 2047. The baseline for zero amplitude is digitally represented as 1024. After baseline adjustment, the range of values becomes $[-1024, 1023]$.

If adaptive quantization has been specified, then we vary the quantization parameter q from a value $q_{\max} = 6$ down to a value $q_{\min}$ till we reach a limit on N_RMSE (20) between $\mathbf{y}$ and $\tilde{\mathbf{y}}$ as specified by the parameter ρ in the stream header.

2) Entropy Model: We model the quantized measurements as samples from a quantized Gaussian distribution that can only take integral values. First, we estimate the mean μ_y and standard deviation σ_y of measurement values in $\bar{\mathbf{y}}$. We round up the values of μ_y and σ_y to the nearest integer for efficient encoding. Entropy coding works with a finite alphabet. Accordingly, the quantized Gaussian model requires specification of the minimum and maximum values that our quantized measurements can take. The range of values in $\bar{\mathbf{y}}$ must be clipped to this range. The clipping function for a scalar value is defined as:

$$\text{clip}(v, a, b) \triangleq \begin{cases} a & v \leq a \\ b & v \geq b \\ v & \text{otherwise.} \end{cases} \tag{7}$$

We clip the values in $\bar{\mathbf{y}}$ to the range $[\mu_y - r\sigma_y, \mu_y + r\sigma_y]$ where r is the range parameter estimated for each frame. Similar to adaptive quantization, we vary r from 2 to 8 till we have captured sufficient variation in $\bar{\mathbf{y}}$ and $\text{N\_RMSE}(\bar{\mathbf{y}}, \hat{\mathbf{y}}) \leq \gamma$ where γ is a parameter specified in the stream header.

The adaptive quantization and adaptive clipping ensure that the total quantization error introduced by quantization and clipping steps is bounded.

3) Entropy Coding: We use the ANS entropy coder to encode $\hat{\mathbf{y}}$ into an array $\mathbf{c}$ of 32-bit integers (called words). This becomes the payload of the frame to be sent to the decoder. The total number of bits in the frame payload is the length of the array n_c times 32. Note that we have encoded and transmitted $\hat{\mathbf{y}}$ and not the unclipped $\bar{\mathbf{y}}$. ANS entropy coding is a lossless encoding scheme. Hence, $\hat{\mathbf{y}}$ will be reproduced faithfully in the decoder if there are no bit errors involved in the transmission [5].

4) Integer Arithmetic: The input to digital compressive sensing is a stream of integers. The sensing process with the sparse binary sensing matrix can be implemented using integer sums and lookup. It is possible to implement the computation of approximate mean and standard deviation using integer arithmetic. We can use the normalized mean square error thresholds for adaptive quantization and clipping steps under integer arithmetic. ANS entropy coding is fully implemented using integer arithmetic. The proposed encoder can be fully implemented using integer arithmetic.

5) Bounded Quantization Noise: N_RMSE values of ρ and γ limit the amount of noise introduced by the quantization and clipping steps respectively. From (22) we can see that $\text{SNR} = -20 \log_{10}(\text{N\_RMSE})$. Our typical values are $\rho = 0.01$ and $\gamma = 0.02$. They correspond to 40 dB of noise for quantization and 34 dB of noise for the clipping step.

6) Encoder Computational Complexity: The encoding process is dominated by the matrix multiplication $\mathbf{y} = \Phi\mathbf{x}$. For a general sensing matrix, this operation is $\mathcal{O}(mn)$. A binary sensing matrix has a total of nd ones. Since d is a small constant, hence the total number of lookup and addition operations is $\mathcal{O}(n)$. This significantly reduces the computational complexity of the encoder.

C. Decoding

Decoding of a frame starts by reading the frame header which provides the frame encoding parameters: $\mu_y, \sigma_y, q, r, n_w, n_c$. The frame header is used for building the quantized Gaussian distribution model for entropy decoding. n_c tells us the number of words ($4n_c$ bytes) to be read from the bitstream for the frame payload. The ANS decoder is used to extract the encoded measurement values $\hat{\mathbf{y}}$ from the frame payload. Inverse quantization and windowing are performed to construct the measurement matrix $\tilde{\mathbf{Y}}$ which is the input to a suitable sparse recovery algorithm.

The architecture is flexible in terms of the choice of the reconstruction algorithm.

$$\tilde{\mathbf{X}} = \text{reconstruct}(\tilde{\mathbf{Y}}). \tag{8}$$

Each column (window) in $\tilde{\mathbf{Y}}$ is decoded independently. In our experiments, we have built two different algorithms:

- BSBL-BO (Block Sparse Bayesian Learning-Bound Optimization) [13], [15], [26]
- CS-NET [16]

Once each window has been reconstructed, they are flattened to form the sequence of reconstructed samples.

D. BSBL-BO

Natural signals tend to have richer structures beyond sparsity. A common structure in natural signals is a block/group structure [27]. We introduce the block/group structure on $\mathbf{x}$ as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_g \end{pmatrix} \tag{9}$$

where each $\mathbf{x}_i$ is a block of b values. The signal $\mathbf{x}$ consists of g such blocks. Under the block sparsity model, only a few $k \ll g$ blocks are nonzero (active) in the signal $\mathbf{x}$ however, the locations of these blocks are unknown. We can rewrite the sensing equation as:

$$\mathbf{y} = \sum_{i=1}^{g} \Phi_i \mathbf{x}_i + \mathbf{e} \tag{10}$$

by splitting the sensing matrix into blocks of columns appropriately. $\mathbf{e}$ denotes the measurement error.

Under the sparse Bayesian framework [15], each block is assumed to satisfy a parametrized multivariate Gaussian distribution:

$$\mathbb{P}(\mathbf{x}_i; \gamma_i, \mathbf{B}_i) = \mathcal{N}(\mathbf{0}, \gamma_i \mathbf{B}_i), \; \forall \; i = 1, \dots, g \tag{11}$$

with the unknown parameters γ_i and $\mathbf{B}_i$. γ_i is a non-negative parameter controlling the block sparsity of $\mathbf{x}$. When the block sparse Bayesian model for $\mathbf{x}$ is estimated, most γ_i tend to be zero due to automatic relevance determination [28] promoting block sparsity. $\mathbf{B}_i \in \mathbb{R}^{b \times b}$ is a positive definite matrix, capturing the correlation structure of the i-th block. We further

[5] We assume that appropriate channel coding mechanism has been used.

assume that the blocks are mutually uncorrelated. The prior of $\mathbf{x}$ can then be written as

$$\mathbb{P}(\mathbf{x}; \{\gamma_i, \mathbf{B}_i\}_i) = \mathcal{N}(\mathbf{0}, \Sigma_0) \tag{12}$$

where

$$\Sigma_0 = \text{diag}\{\gamma_1 \mathbf{B}_1, \dots, \gamma_g \mathbf{B}_g\}. \tag{13}$$

We also model the correlation among the values within each active block as an AR-1 process with a common model parameter. Under this assumption the matrices $\mathbf{B}_i$ take the form of a Toeplitz matrix

$$\mathbf{B}_i = \mathbf{B} = \begin{bmatrix} 1 & r & \dots & r^{b-1} \\ r & 1 & \dots & r^{b-2} \\ \vdots & & \ddots & \vdots \\ r^{b-1} & r^{b-2} & \dots & 1 \end{bmatrix} \tag{14}$$

where r is the AR-1 model coefficient. This constraint significantly reduces the number of model parameters to be learned.

Measurement error is modeled as an independent zero-mean Gaussian noise $\mathbb{P}(\mathbf{e}; \lambda) \sim \mathcal{N}(\mathbf{0}, \lambda \mathbf{I})$. BSBL doesn't require us to provide the value of noise variance as input. It is able to estimate λ within the algorithm. The estimate of $\mathbf{x}$ under the Bayesian learning framework is given by the posterior mean of $\mathbf{x}$ given the measurements $\mathbf{y}$.

Our implementation of the BSBL-BO algorithm is available as part of CR-Sparse library [29]. As Zhang et al. suggest in [13], block sizes are user-defined and they are identical and no pruning of blocks is applied. Our implementation has been done under these assumptions and is built using JAX so that it can be run on GPU hardware easily to speed up decoding. The only configurable parameter for this decoder is the block size which we shall denote by b in the following.

E. CSNet

CSNet [16] is a state-of-the-art deep learning network for the reconstruction of ECG signals from compressive measurements. The measurements are first raised back to $\mathbb{R}^n$ as $\mathbf{z} = \Phi^T \mathbf{x}$. An initial reconstruction module consists of three convolution layers. It is followed by the secondary reconstruction module consisting of an LSTM layer followed by a dense layer. We implemented this network and followed the training procedure as described in [16]. Our primary change was that we didn't feed the original measurements. Rather, we trained the network with $\tilde{\mathbf{Y}}$ as inputs and $\mathbf{X}$ as expected outputs. Thus, we tested whether CSNet can work well with our quantized and clipped measurements. Another difference from [16] is that we use sparse binary sensing matrices rather than standard RIP compliant sensing matrices.

III. ECG Database and Performance Metrics

We use the MIT-BIH Arrhythmia Database [30] from PhysioNet [31]. The database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings from 47 subjects. The recordings were digitized at 360 samples per second for both channels with 11-bit resolution over a 10mV range. The samples can be read in both digital (integer) form or as physical values (floating point) via the software provided by PhysioNet. We use the MLII signal (first channel) from each recording in our experiments.

Each window of n samples generates m measurements by the sensing equation $\mathbf{y} = \Phi \mathbf{x}$. Assume that we are encoding s ECG samples where $s = nw$ and w is the number of signal windows being encoded (across all frames). Let the ECG signal be sampled by the ADC device at a resolution of r bits per sample [6]. Then the number of uncompressed bits is given by $\text{bits}_u = rs$. Let the total number of compressed bits corresponding to the s ECG samples be bits_c [7]. Then the *compression ratio* (CR) is defined as

$$\text{CR} \triangleq \frac{\text{bits}_u}{\text{bits}_c}. \tag{15}$$

Percentage space saving (PSS) is defined as

$$\text{PSS} \triangleq \frac{\text{bits}_u - \text{bits}_c}{\text{bits}_u} \times 100. \tag{16}$$

We call the ratio m/n as the *measurement ratio*. The *percentage measurement saving* (PMS) is defined as:

$$\text{PMS} \triangleq \frac{n - m}{n} \times 100. \tag{17}$$

Another way of looking at the compressibility is how many bits per sample (bps) are needed on average in the compressed bitstream. We define bps as:

$$\text{bps} \triangleq \frac{\text{bits}_c}{s}. \tag{18}$$

Similarly, we can define *bits per measurement* (bpm) as:

$$\text{bpm} \triangleq \frac{\text{bits}_c}{mw}. \tag{19}$$

The *normalized root mean square error* is defined as

$$\text{N\_RMSE}(\mathbf{x}, \tilde{\mathbf{x}}) \triangleq \frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \tag{20}$$

where $\mathbf{x}$ is the original ECG signal and $\tilde{\mathbf{x}}$ is the reconstructed signal. A popular metric to measure the quality of reconstruction of ECG signals is *percentage root mean square difference* (PRD):

$$\text{PRD}(\mathbf{x}, \tilde{\mathbf{x}}) \triangleq \text{N\_RMSE}(\mathbf{x}, \tilde{\mathbf{x}}) \times 100 \tag{21}$$

The *signal to noise ratio* (SNR) is related to PRD as

$$\text{SNR} \triangleq -20 \log_{10}(0.01 \text{PRD}). \tag{22}$$

Zigel et al. [32] established a link between the diagnostic distortion and the easy to measure PRD metric. Table III shows the classified quality and corresponding SNR and PRD ranges.

IV. Experiments and Discussion

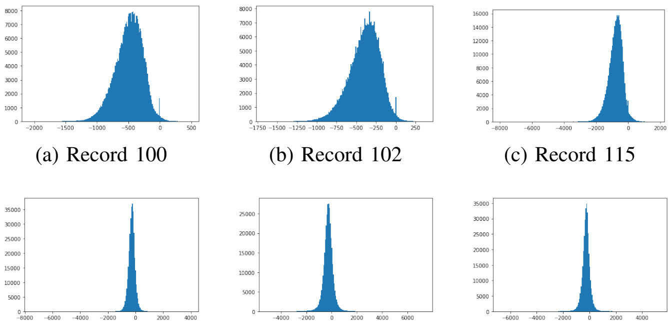
The experiments in this section have been designed to study the compression efficiency of the encoder and the quality of reconstruction under different reconstruction algorithms.

[6] For MIT-BIH Arrhythmia database, $r = 11$.

[7] This includes the overhead bits required for the stream header and frame headers to be explained later.

TABLE III
QUALITY OF RECONSTRUCTION [32]

Quality	PRD	SNR
Very good	$< 2\%$	> 33 dB
Good	2-9%	20-33 dB
Undetermined	$\geq 9\%$	≤ 20 dB



(a) Record 100 (b) Record 102 (c) Record 115

(d) Record 202 (e) Record 208 (f) Record 234

Fig. 3. Histograms of measurement values with the sparse binary sensing matrix with $m = 256, n = 512, d = 4$ in 200 bins

A. Gaussianity

The key idea behind our entropy coding design is to model the measurement values as being sampled from a quantized Gaussian distribution. Figure 3 shows the histograms of measurement values for 6 different records. Visually, the quantized Gaussian distribution approximation appears to be quite reasonable.

The best compression can be achieved by using the empirical probabilities of different values in $\mathbf{y}$ in an entropy model. However, doing so would require us to transmit the empirical probabilities as side information. This may be expensive. We can estimate the loss in compression overhead by use of the quantized Gaussian approximation. Let $\mathbb{P}$ denote the empirical probability distribution of data and let $\mathbb{Q}$ denote the corresponding quantized Gaussian distribution. Bamler in [20] show empirically that the overhead of using an approximation distribution $\mathbb{Q}$ in place of $\mathbb{P}$ in ANS entropy coding is close to the KL divergence $\text{KL}(\mathbb{P}||\mathbb{Q})$ which is given by

$$\text{KL}(\mathbb{P}||\mathbb{Q}) = \sum_{y} \mathbb{P}(y) \log_2 \left(\frac{\mathbb{P}(y)}{\mathbb{Q}(y)} \right). \tag{23}$$

We computed the empirical distribution for $\mathbf{y}$ for each record and measured its KL divergence with the corresponding quantized Gaussian distribution. It varies around 0.11 ± 0.07 bits across the 48 records for an encoder configuration with $m = 256, n = 512, d = 4$. Thus, the overhead of using a quantized Gaussian distribution in place of the empirical probabilities can be estimated to be $4 - 18\%$. The empirical distributions vary widely from one record to another in the database. Hence using a single fixed empirical distribution (e.g. the Huffman codebook preloaded into the device in [10]) may lead to lower compression. The simple adaptive quantization approach proposed here provides an effective alternative yielding superior performance.

B. Quantization Parameter

The quantization step is a key contributor to compression savings. This can be studied better under the fixed quantization mode. In the first experiment, we compress the data from record 100 at different values of the quantization parameter q (6) and PMS. The results are shown in fig. 4. The number of measurements (m) has been chosen to vary from 20% to 60% of the window size (n). q varies from 0 to 7. Figure 4 (a) shows that reconstruction quality doesn't change much from $q = 0$ till $q = 4$ after which it starts degrading. As expected, the PRD degrades as m is reduced. (b) shows that PSS (percentage space saving) increases linearly with q. As q increases, the size of the alphabet for entropy coding reduces and this leads to increased space savings. (c) shows the variation of PSS with PMS at different values of q. PSS increases linearly with PMS. Also, PSS is much higher than PMS at higher values of q.

Increasing quantization linearly increases the PSS. Since up to $q = 4$, there is no noticeable impact on PRD, as seen in fig. 4 (a), it is safe to enjoy these savings in bit rate. At 40% PMS, one can attain up to 69.3% PSS without losing any reconstruction quality.

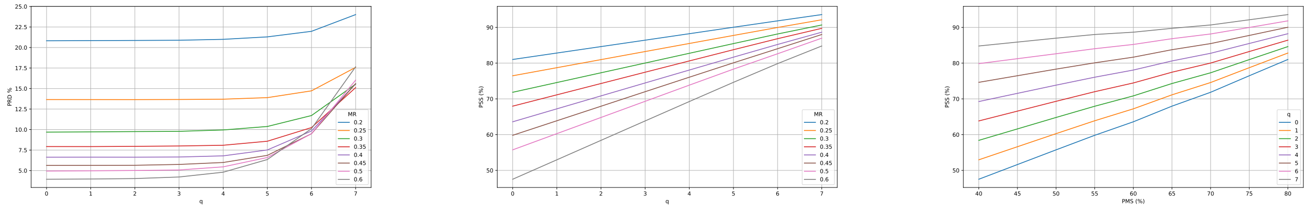
Figure 5 shows the impact of the quantization step on the reconstruction quality for a small segment of record 100. The first panel shows the original (digital) signal. The remaining panels show the reconstruction at different q values using the BSBL-BO algorithm. The reconstruction visual quality is excellent up to $q = 5$ (PRD below 7%), good at $q = 6$ (PRD at 9%) and clinically unacceptable at $q = 7$ (with PRD more than 15%). One can see significant waveform distortions at $q = 7$. Also, note how the quality score keeps increasing till $q = 4$ and starts decreasing after that with a massive drop at $q = 7$.

In the remaining experiments, we will be using adaptive quantization.

We note that Mamaghanian et al. [10] have used an inter-packet redundancy removal step before Huffman coding. They suggest that measurements of consecutive windows are correlated due to the periodic nature of the ECG signal. However, we suspect that this was happening due to the use of unsigned digital values with a large DC component in their design. Since we removed the baseline from the signal before compression, we didn't notice any significant inter-packet redundancy. Hence, we haven't included any redundancy removal block in our encoder.

C. Space Savings

In this experiment, we report the variation of the percentage space savings (PSS) with the percentage measurement savings (PMS) under different encoder configurations. We chose the window size $n = 512$. PMS varies from 20% to 80%. Correspondingly m varies from 410 down to 77. For each choice of m and n, we constructed binary sensing matrices Φ for two different values of d at $d = 4$ and $d = 12$. All 48 records were encoded and PSS was measured separately for each of them. Figure 6 shows the variation of mean PSS (over records) with PMS along with error bars for standard deviation across records at each PMS. We note that $d = 12$



(a) q vs PRD

(b) q vs PSS

(c) PMS vs PSS

Fig. 4. Variation of compression statistics vs quantization parameter at different measurement ratios $\frac{m}{n}$ for record 100

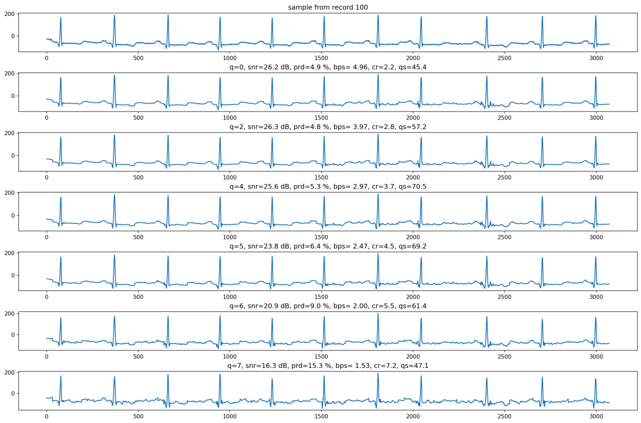


Fig. 5. Reconstruction of a small segment of record 100 for different values of $q = 0, 2, 4, 5, 6, 7$ with $m = 256, n = 512, d = 4$ under non-adaptive quantization. The block size for the BSBL-BO decoder is 32.

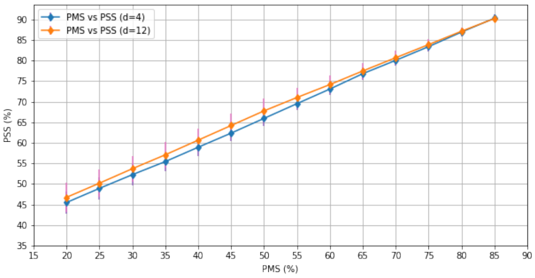


Fig. 6. Error bars for variation of mean PSS with PMS for 48 records at $d = 4$ and $d = 12$.

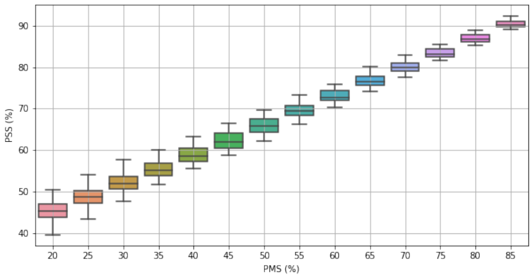


Fig. 7. PMS vs PSS box plots over 48 records at $d = 4$

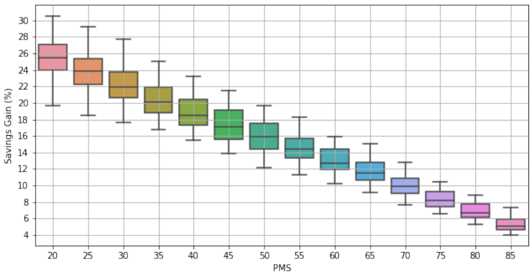


Fig. 8. PMS vs Savings gain (PSS - PMS) box plots over 48 records at $d = 4$

gives slightly better compression. However, the difference is not significant and it decreases as PMS increases. The small error bars indicate that the compression ratio doesn't vary much from record to record. At low PMS, we can see that the compression scheme can give additional savings of up to 25%. The trend line is linear and savings reduce linearly to 5% at 85% PMS.

Figure 7 shows more detailed box plots of variation of PMS across the 48 records at different PMS values.

In the following, we will report results for $d = 4$ configuration. Results are similar for $d = 12$.

We measured the overhead of bits required to encode the stream and frame headers. The overhead varies from 0.07% at PMS=20% to about 0.4% at PMS=80% on average. At higher PMS there are very few measurements to encode. Hence overhead is higher. Increasing the frame size will reduce the overhead. However, this will cause delays in a real-time system since a frame cannot be decoded till its full payload has been received.

The bits per sample vary from 6 ± 0.3 bps at PMS=20% down to 1 ± 0.1 bps at PMS=85%. This is significant compared to the uncompressed rate of 11 bits per sample.

Savings gain can be measured as the difference $\text{PSS} - \text{PMS}$. Figure 8 shows the box plots of savings gains at different PMS levels. Savings gain varies from $25 \pm 3\%$ at PMS=20% down to $5.3 \pm 0.9\%$ at PMS=85%. More measurements lead to more savings via the quantization, clipping and entropy coding steps.

Our adaptive quantization scheme is designed to keep the noise introduced by quantization and clipping steps to reasonable levels. Figure 9 shows the box plots of variation of quantization noise SNR across the 48 records at different PMS. It is clear that quantization SNR remains limited between 35 and 40 dB.

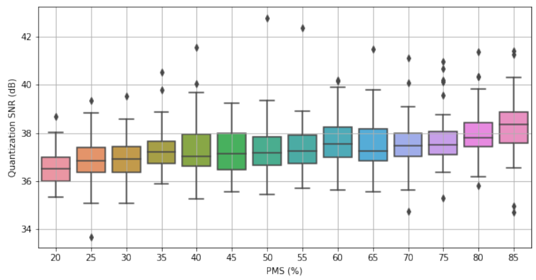


Fig. 9. PMS vs Quantization SNR box plots over 48 records at $d = 4$

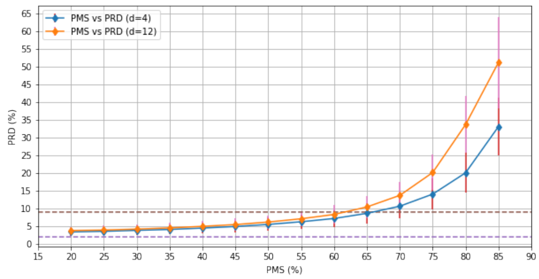


Fig. 10. Error bars for variation of PMS vs mean PRD at $d = 4$ and $d = 12$ for reconstruction with the BSBL-BO algorithm.

Compression is to no avail if the signal cannot be reconstructed faithfully. The reconstruction heavily depends on the choice of the sparse recovery algorithm. In the following, we present our results for two different sparse recovery approaches.

D. Reconstruction with BSBL-BO

Figure 10 shows the variation of mean PRD with PMS for $d = 4$ as well as $d = 12$ when BSBL-BO reconstruction algorithm is used in the decoder. The error bars show the standard deviation of PRD across the 48 records. We have drawn additional lines for 2% and 9% PRD that indicate very good and good reconstruction levels. Surprisingly, $d = 4$ gives us better PRD. We can see that mean PRD is below 9% level till up to 65% PMS. At this PMS, we have a PSS of about 77% (fig. 6).

Figure 11 provides more detailed box plots for the variation of PRD with PMS across the 48 records at $d = 4$.

E. Reconstruction with CSNet

For CSNet, a window size of $n = 256$ was chosen following [16]. We only trained and tested for a fixed binary sensing matrix with $d = 4$. Figure 12 shows detailed box plots for the variation of PRD with PMS across the 48 records. We note that CSNet can reconstruct well up to 80% PMS with 7% additional space savings (mean PSS being 87%). Reconstruction quality degrades far more slowly for CSNet with mean PRD being 16.5 % for a PMS of 90 % and mean PSS being 93.4%.

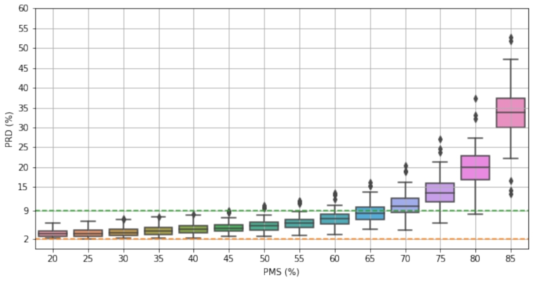


Fig. 11. PMS vs PRD box plots over 48 records at $d = 4$ for reconstruction with BSBL-BO algorithm

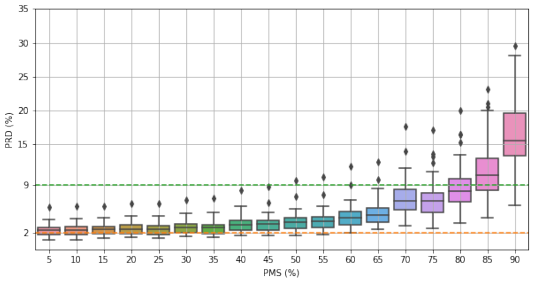


Fig. 12. PMS vs PRD box plots over 48 records at $d = 4$ for reconstruction with CS-NET

F. Comparison

Zhang et al. [16] have reported optimal PMS values for different methods (BP, OMP, BSBL-BO, CSNet, etc.) for PRD=9% for a subset of test records for $n = 256$. In their setup, Φ is RIP satisfying matrix which is different from the binary sensing matrix used in our setup. For BSBL-BO reconstruction, we report the needed PMS as well as the corresponding PSS for these records for a PRD $\leq 9\%$. in table IV. While our codec requires slightly more measurements (lower PMS) due to the sparsity of the sensing matrix, it consistently outperforms in compression due to additional space savings from quantization and entropy coding. Our PSS is better by $6.5 \pm 3\%$.

TABLE IV
REQUIRED PMS AND CORRESPONDING PSS FOR BSBL-BO FOR A TARGET PRD $\leq 9\%$

Record	Ref PMS	Our PMS	PSS	PRD
100	71	66	78.0	8.8
101	71	66	77.2	8.9
102	70	61	75.8	8.7
107	74	65	75.2	8.9
109	75	70	78.8	8.7
111	69	63	74.5	8.6
115	68	67	78.4	8.7
117	74	73	83.9	7.2
118	76	70	81.3	8.2
119	71	70	81.1	7.7

Similarly, in table V, we compare the PMS reported in [16] for CSNet for good recovery with the needed PMS as well as the corresponding PSS for these records in our CSNet implementation for sparse binary sensing matrices. Again, we see slightly lower PMS (i.e., more measurements required) to attain the target PRD but this is more than compensated for by the higher PSS levels. We reiterate that studies like [16] simply assume that the measurements also require the same 11-bit representation like original data and ignore the issue of entropy coding.

TABLE V
REQUIRED PMS AND CORRESPONDING PSS FOR CSNET FOR A TARGET PRD $\leq 9\%$

Record	Ref PMS	Our PMS	PSS	PRD
100	88	80	87.8	8.6
101	88	79	86.8	8.8
102	63	56	72.8	8.2
107	81	78	83.3	8.4
109	87	84	89.4	8.7
111	80	78	84.4	8.5
115	91	88	92.3	7.6
117	93	92	95.2	8.7
118	87	84	91.0	8.9
119	92	91	94.0	8.7

Mamaghanian et al. [10] report their best result for CS for record 107 for which they report that ”good” (PRD $\leq$ 9%) signal recovery can be achieved up to a PSS of 74% using SPGL1 reconstruction. For our codec with BSBL-BO reconstruction, good recovery was achieved up to a PSS of 75.2 %. With CSNet, it increases further up to 83.3 %.

V. CONCLUSION

We summarize the key features and benefits of our encoding scheme. Our scheme converts digital compressive measurements to a finite alphabet suitable for entropy coding using simple adaptive quantization and clipping steps with limited quantization noise. The quantized measurements can be approximately modeled using a quantized Gaussian distribution for entropy coding. ANS entropy coding can efficiently encode them into a bitstream. The encoder can be easily implemented using integer arithmetic on resource-limited devices. Our entropy coding scheme does not require a fixed codebook. It can adapt to changing signal characteristics frame by frame. We see additional PSS gains of 5-10% (at higher PMS and PRD) up to 25% (at lower PMS and PRD). The decoder can use any suitable reconstruction algorithm. We have shown it working with BSBL-BO and CSNet. The codec architecture is general enough to be applied in other compressive sensing applications also.

The software code implementing this codec and all scripts for the experimental studies conducted in this work have been released as opensource software on GitHub [33].

While the quantized Gaussian entropy model has worked well in this design, better entropy models require further research. Lower values of d tend to give better SNR with less number of measurements. It is unclear why this happens. Surprisingly, changing d doesn't affect the compression ratio much. These issues require further investigation. More work is required in applying this architecture to other physiological signals. Images are also a candidate for this architecture. The architecture will need to be modified to handle multidimensional data.

ACKNOWLEDGMENT

Shailesh Kumar would like to thank his employer Interra Systems Inc. for allowing him to pursue his research interests along with his regular job.

REFERENCES

[1] H. Cao, V. Leung, C. Chow, and H. Chan, “Enabling technologies for wireless body area networks: A survey and outlook,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009.

[2] A. Milenković, C. Otto, and E. Jovanov, “Wireless sensor networks for personal health monitoring: Issues and an implementation,” *Computer communications*, vol. 29, no. 13-14, pp. 2521–2533, 2006.

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

[4] R. G. Baraniuk, “Compressive sensing [lecture notes],” *Signal Processing Magazine, IEEE*, vol. 24, no. 4, pp. 118–121, 2007.

[5] E. J. Candès, “Compressive sampling,” in *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

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

[7] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.

[8] 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, 2014.

[9] S. S. Kumar and P. Ramachandran, “Review on compressive sensing algorithms for ecg signal for iot based deep learning framework,” *Applied Sciences*, vol. 12, no. 16, p. 8368, 2022.

[10] 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.

[11] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, “Compressed sensing based method for ecg compression,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 761–764.

[12] J. Zhang, Z. Gu, Z. L. Yu, and Y. Li, “Energy-efficient ecg compression on wireless biosensors via minimal coherence sensing and weighted l1 minimization reconstruction,” *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 520–528, 2014.

[13] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, “Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ecg via block sparse bayesian learning,” *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, 2012.

[14] Z. Zhang, T.-P. Jung, S. Makeig, Z. Pi, and B. D. Rao, “Spatiotemporal sparse bayesian learning with applications to compressed sensing of multichannel physiological signals,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 22, no. 6, pp. 1186–1197, 2014.

[15] Z. Zhang and B. D. Rao, “Extension of sbl algorithms for the recovery of block sparse signals with intra-block correlation,” *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009–2015, 2013.

[16] H. Zhang, Z. Dong, Z. Wang, L. Guo, and Z. Wang, “Csnet: A deep learning approach for ecg compressed sensing,” *Biomedical Signal Processing and Control*, vol. 70, p. 103065, 2021.

[17] K. Luo, J. Li, and J. Wu, “A dynamic compression scheme for energy-efficient real-time wireless electrocardiogram biosensors,” *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 9, pp. 2160–2169, 2014.

[18] S. A. Chouakri, O. Djaafri, and A. Taleb-Ahmed, “Wavelet transform and huffman coding based electrocardiogram compression algorithm: Application to telecardiology,” in *Journal of Physics: Conference Series*, vol. 454, no. 1. IOP Publishing, 2013, p. 012086.

[19] J. Duda, “Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding,” *arXiv preprint arXiv:1311.2540*, 2013.

[20] R. Bamler, "Understanding entropy coding with asymmetric numeral systems (ans): a statistician's perspective," *arXiv preprint arXiv:2201.01741*, 2022.

[21] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[22] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes rendus mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[23] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 798–805.

[24] R. Berinde, P. Indyk, and M. Ruzic, "Practical near-optimal sparse recovery in the l1 norm," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 198–205.

[25] E. Van Den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *Siam journal on scientific computing*, vol. 31, no. 2, pp. 890–912, 2009.

[26] Z. Zhang, S. Wei, D. Wei, L. Li, F. Liu, and C. Liu, "Comparison of four recovery algorithms used in compressed sensing for ecg signal processing," in *2016 Computing in Cardiology Conference (CinC)*. IEEE, 2016, pp. 401–404.

[27] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[28] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[29] S. Kumar, "Cr-sparse: Hardware accelerated functional algorithms for sparse signal processing in python using jax," *Journal of Open Source Software*, vol. 6, no. 68, p. 3917, 2021.

[30] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[31] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

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

[33] Kumar, Shailesh, "Compressive sensing based ecg codec," 2022. [Online]. Available: https://github.com/shailesh1729/ecg-codec