Boosting the Battery Life of Wearables for Health Monitoring through the Compression of Biosignals

Mohsen Hooshmand, Davide Zordan, Davide Del Testa, Enrico Grisan, Michele Rossi

***Abstract*—Modern wearable IoT devices enable the monitoring of vital parameters such as heart or respiratory rates (RESP), electrocardiography (ECG), photo-plethysmographic (PPG) signals within e-health applications. However, a common issue of wearable technology is that signal transmission is power-demanding and, as such, devices require frequent battery charges and this poses serious limitations to the continuous monitoring of vitals. To ameliorate this, we advocate the use of lossy signal compression as a means to decrease the data size of the gathered biosignals and, in turn, boost the battery life of wearables and allow for fine-grained and long-term monitoring. Considering one dimensional biosignals such as ECG, RESP and PPG, which are often available from commercial wearable IoT devices, we provide a throughout review of existing biosignal compression algorithms and introduce novel approaches based on online dictionaries, elucidating their operating principles and providing a quantitative assessment of their compression, reconstruction and energy consumption performance. As we quantify, the most efficient schemes allow reductions in the signal size of up to 100 times, which entail similar reductions in the energy demand, by still keeping the reconstruction error within 4% of the peak-to-peak signal amplitude. Avenues for future research are finally discussed.**

***Index Terms*—wearable IoT devices, biomedical signal processing, signal compression, sparse autoencoders, pattern recognition, energy efficiency.**

I. Introduction

INTERNET of Things (IoT) technology enables objects to sense the physical environment and to seamlessly integrate the gathered data into sophisticated Internet applications that allow for substantial improvements of human activities at large. The focus of this paper is on human sensing [1] through wearable IoT devices, such as smart watches, chest straps or wristbands, which can be used to help address the individual health and the fitness needs of the users [2]. For instance, wearables can be utilized to gather and share information about the status of outpatients, making it possible to collect, record and analyze new data streams faster and more accurately. This allows for an improved access to healthcare, an increase of its quality and ultimately, a reduction in its cost. Telehealth systems could deliver care to people in remote locations and provide streams of accurate data for making better care decisions (e.g., in terms of therapy adjustments or prompt interventions). In addition, these systems are expected to have a big impact on the field of rehabilitation, where, e.g., users may wear e-textile systems for remote, continuous monitoring of physiological and movement data [3]. Through IoT technology, a large number of physiological signals can be monitored including oxygen saturation, blood pressure, heart rate, respiration rate, glucose level [2], [4] and user activities such as walking, standing, sleeping, etc., can be inferred [5]. A recent survey of wearable devices and their use is offered in [2], whereas rehabilitation systems are discussed in [3].

We look at an IoT scenario for e-health, where wearables are utilized to collect physiological signals, preprocess and transmit them over their wireless interface for their final storage and manipulation via backend server infrastructures. Within this context, we are concerned with the design of online signal compression algorithms, so that the gathered signals can be effectively stored in the limited memory space of wearables and conveniently transmitted over their radio interface. Ideally, we would like this software to adapt to the signals being sampled, by being prompt when required by the application and gently go into some power saving mode when the signals exhibit regular patterns. This means that, high resolution should be provided when the user is up to some dynamic activity and wants to track that or when a critical behavior is detected. Toward this end, we advocate the use of lossy compression as a means to reduce the space taken by the collected biosignals and, at the same time, to save battery power through a reduced transmission time. This amounts to compressing the physiological data directly at its source.

As for the physiological signals of interest, we consider one dimensional and quasi-periodic biomedical signals as those provided by typical sensors in chest straps or wristbands, i.e., electrocardiography (ECG), photo-plethysmographic (PPG) and respiratory (RESP) signals. ECG is probably the most important among them for the diagnosis of heart malfunctions and IoT technologies are expected to be very useful to assess cardiac conditions within patient-monitoring applications. Commercial devices such as the Bioharness 3 from Zephyr Technology Corporation [6] can be utilized to measure this type of signal. RESP signals are also very relevant and can be obtained from chest straps [7] or rubber straps [8] placed around the abdomen to, e.g., assess the status of outpatients affected by chronic respiratory failure and allow monitoring them from home. PPG is often available in low-cost IoT devices for the consumer market (such as smart watches or wristbands designed for fitness applications), see the Angel sensor wristband [9]. PPG can be used to estimate heart-rate [10] and recent studies indicate that blood pressure can also be inferred [11].

We believe that, despite the focus and hype on wearable technology, research on data processing algorithms for wearable IoT devices is still in its infancy and most still has to be done to take full advantage of this portable technology, especially in the medical field. In past research, a large number of compression algorithms were proposed for ECG,

All authors are with the department of Information Engineering (DEI) at the University of Padova, via Gradenigo 6/b, 35131, Padova, Italy.

but signal compression has never been applied to RESP or PPG. Moreover, performance assessments were only carried out for quality of compression and reconstruction, whereas the energy consumption aspect has often been neglected. Instead, we stress that energy should be sparingly used by the software running on wearables, as these devices are often battery operated and, in turn, their energy consumption is a key consideration. Also, to the best of the authors' knowledge, no quantitative comparison among existing solutions can be found in the literature and, due to this, it is unclear which algorithms are best suited for use in wearable devices.

In this paper, we aim at filling these gaps. First, in Section II we present a taxonomy of popular signal compression schemes from the literature, touching upon linear approximations [12], [13], Fourier [14], Wavelet [15] transforms and novel compression techniques based on *compressive sensing* [16], [17] and *denoising autoencoders* [18].

A novel compression architecture based on vector quantization and pattern recognition [19] is proposed in Section III-A, where a suitable codebook (or dictionary) is built and maintained in an online fashion to efficiently represent data patterns. Compression is achieved as codebook indices are sent to the decompressor in place of the original time series. Despite its simplicity, this technique is found to be appealing due to its excellent performance in the high compression regime. The other selected algorithms from the literature are detailed in Section III and a comparative performance evaluation of all the considered compression approaches is carried out in Section IV, where we quantify their compression efficiency, signal reconstruction fidelity and, most importantly, their energy consumption. Also, we estimate the battery time improvement due to the adoption of the discussed compression technology for continuous monitoring applications.

Finally, our conclusions are presented in Section V, along with a discussion of open research issues.

To summarize, the main contributions of this paper are:

- A taxonomy of existing signal compression schemes that are amenable to implementation on wireless wearable IoT devices.
- A simple but effective dictionary-based approach to the *online* classification and compression of biosignals, along with its validation.
- A detailed performance evaluation of the considered compression schemes in terms of reconstruction error, energy consumption (isolating the energy required for compression and transmission) and compression efficiency when applied to ECG, RESP and PPG signals.
- A discussion of open areas for improvement and new research avenues.

II. Taxonomy of Lossy Compression Schemes

In the last few years, a great deal of work has been carried out on tools for the efficient ECG signal analysis, facial image recognition or the identification of fingerprints acquired by a cell phone, see [20]. PPG is being intensively investigated for the estimation of the heart rate [10] and motion data is being used for activity detection [21]. Nevertheless, apart from ECG, little has been done regarding the compression of other signals, such as PPG, RESP, etc. In this taxonomy, we first focus on ECG and then elaborate on the use of compression for other signal types.

The two most important tasks to be accomplished in the ECG domain are 1) QRS complex detection and 2) signal compression. As per QRS detection, it is crucial to split the ECG time series into heart beat segments (one segment per beat) as this allows the fine-grained assessment of inter-beat signal features, which are useful to detect certain pathologies. Note that ECG can be efficiently split into beat segments as it is a quasi-periodic time series exhibiting recurrent patterns. As per signal compression, we emphasize that wearable devices are energy and memory constrained and, as such, minimizing the amount of data to store and send is an important consideration. As an example, a typical sampling rate of 250 samples per second with 12 bits per sample (e.g., from a Zephyr's Bioharness device) leads to 32.4 Mbytes of data for a full day. As we will see below, compression algorithms can easily reduce this number by 70 times to about 463 kbytes, leading to much higher efficiencies in terms of memory and transmission.

1) QRS complex detection has been extensively studied in the literature. Several methods were proposed to detect QRS complexes and to enhance their features. The importance of QRS enhancement has been demonstrated to detect the QRS complex [22]. In particular, amplitude thresholding [23], first and second derivative methods [24], mathematical morphology [25], [26], filter banks [27], and wavelet transform techniques [28] are among the methods used for the enhancement of the QRS complex. The QRS detection is instead usually performed with a combination of techniques such as thresholding [23], [25], neural networks [29], wavelet transform [30], matched filters [31]. These techniques are of foremost importance as they split the ECG time series into segments (i.e., the data points between subsequent heartbeats), which are then utilized for the subsequent estimation of the pulse, and for the compression of the ECG trace.

2) Signal compression. Quite a few lossy and lossless compression algorithms for ECG signals have been proposed in the literature in the last decades. Typically, they can be classified into three main categories:

- **Time domain processing**: within this class we have AZTEC [32], CORTES [33] and Lightweight Temporal Compression (LTC) [12]. AZTEC and CORTES achieve compression by discarding some of the signal samples and applying a linear approximation, whereas LTC approximates the original time series through piecewise linear segments, where the two end points of a segment are sent in place of the points in between. As we show in Section IV, in spite of its simplicity, LTC closely matches the performance of Principal Component Analysis (PCA) [13], [34].
- **Transform based coding**: these exploit transformations such as Fast Fourier Transform (FFT) [14], Discrete

Cosine Transform (DCT) [35] and Discrete Wavelet Transform (DWT) [15]. The rationale behind them is to represent the signal in a suitable transform domain and select a number of transform coefficients to be sent in place of the original samples. The amount of compression depends on the number of coefficients that are selected, the representation accuracy depends on how many and which coefficients are retained. Although the schemes belonging to this class have good compression capabilities, their computational complexity is often too high for wearable devices [36]. Lightweight implementations are possible and are considered in the present paper. However, simpler linear and dictionary based algorithms have better performance in terms of reconstruction error as we show in Section IV.

- **Parametric techniques**: these schemes use neural networks [37], vector quantization [38], Compressed Sensing (CS) [16] and pattern matching [39]. Their rationale is to process the temporal series to obtain some kind of knowledge and use it to predict the signal behavior. Recently, denoising autoencoders [18] have been proposed as universal approximators of biosignal patterns and have been shown to provide excellent compression performance and to have much smaller computational costs than competing algorithms. This is a field with limited investigation up to now. Also, these algorithms have promising capabilities for the extraction of signal features.

Despite these developments, we recall that no systematic comparison was carried out in the existing literature and, more than that, the proposed algorithms were not evaluated in terms of their energy expenditure. This is of course very important for wearables, which are battery operated and thus call for algorithms that are at the same time extremely effective and computationally cheap.

In addition, besides ECG, recent advances in technology for wearable devices have made it possible to efficiently collect and analyze other signals such as PPG, motion and respiration through body worn sensor technologies [40]. The PPG signal can be a powerful diagnostic tool due to simple, portable, and low-cost technology available for its fast, easy, and reliable acquisition and can be non-intrusively measured using wristbands or smart-watches. An increasing number of works in the literature deal with the extraction of physiological parameters from the PPG signal such as heart rate, blood pressure, blood oxigen saturation, and respiration [11], [41], [42]. Nevertheless, to the best of our knowledge no algorithms have been proposed so far for the compression of these signals. Note that with future application developments, besides the calculation of selected features or health indicators right on the mobile devices, users or doctors may want to fully monitor the vitals, which could be sent to smartphones or control centers for further elaboration so as to provide a fine-grained assessment of the patient's condition, e.g., to assess the evolution or occurrence of a certain pathology. In this case, compressed but accurate representations of vital signals from heterogeneous sensor technology are expected to be very useful.

III. Signal Compression Algorithms

Next, we detail the selected signal compression algorithms for quasi-periodic biosignals, by first presenting a novel technique based on the online construction of a dictionary to represent input patterns. The compression methods that we describe below are based on differing paradigms. In fact, some use the degree of similarity (correlation) across subsequent patterns (or segments), whereas others consider the correlation within the same segment. We refer to the former approach to as "inter-segment correlation" based compression, whereas for the latter we use the term "intra-segment correlation". The algorithms belonging to the inter-segment class are: online dictionary, vector quantization and autoencoders, whereas algorithms based on principal component analysis, LTC, discrete cosine and wavelet transforms exploit intra-segment correlation properties. The implementation of compressive sensing that is considered in this paper belongs to both classes.

A. Online Dictionary (OD)

In this section, we propose a dictionary based compression algorithm based on the concept of *motif extraction* [43] and pattern recognition. Its building blocks are shown in Fig. 1 and explained in what follows. Although the scheme is simple (it consists of a single pass vector quantization without codeword reclustering) it provides excellent performance in the high compression regime and its analysis sheds some light on the desirable properties that a compression scheme should have, allowing the assessment of the pitfalls of offline dictionary based schemes and the identification of future research directions, as we discuss in Sections IV and V.

The algorithm belongs to the inter-segment correlation class and can be applied to the biomedical signals exhibiting recurrent patterns such as ECG, photo-plethysmographic traces (PPG), arterial blood pressure (ABP), respiratory signals (RESP), etc. The idea is that recurrent patterns can be efficiently identified and used to construct, *at runtime*, a codebook (also referred to as dictionary). This codebook is built and maintained by the compressor at the transmitter side and has to be synchronized with that at the decompressor at the receiver. The compression of biosignals is achieved by sending, for each input pattern, the corresponding index in the codebook, in place of the original data points. We achieved this through several processing functions, as shown in Fig. 1, namely: 1) a passband filter, 2) a peak detector, 3) a segment extractor, 4) pattern matching and 5) a codebook manager.

1) Passband filtering: as a first step, we use a passband filter to remove artifacts such as high frequency noise and the DC component. For ECG, this filter operates in the band $[8, 20]$ Hz, although these can be changed to best suit other signal types. Here, we implemented the third-order Butterworth filter of [44].

2) Peak detection: with this algorithm we detect the position of the main peaks in the time series. For ECG, these

correspond to the heart beats. To this end, we have adopted the technique of [45], which has been conceived for ECG signals but can be easily modified to effectively work with PPG or respiratory traces. This technique is self-tuning and optimizes itself based on the input data sampling rate. We considered this scheme as it is fast and lightweight and thus suitable for use in wearable and energy constrained devices.

3) Segment extractor: once the peaks are detected, we consider the data samples between subsequent peaks. These constitute the input *segments* for our compressor algorithm. Note that, unlike the common practice of positioning the segments so that the peaks (heart beats) are in their center, we define a segment as the data points *between subsequent peaks*. Hence, all segments are normalized according to a predefined length of W samples, which is the same size of the codewords in the dictionary. This is accomplished by re-stretching the segment length to W samples through interpolation (this block is referred to as "period normalization" in Fig. 1). While in principle any interpolation technique can be used, such as quadratic or spline based, in our implementation we utilized a simple linear technique as we found it sufficiently accurate while also being computationally inexpensive. Working with such segments allows using machine learning algorithms for the construction of the codebook, as we detail shortly.

4) Pattern matching: this block takes the current input segment and checks whether this matches one of the codewords in the codebook (dictionary), which is built and maintained at runtime as we explain in point 5) below. Several matching criteria are possible. One of such criteria may be Dynamic Time Warping (DTW) [46], which has been extensively and successfully used in the literature to compare patterns of different length and can also be implemented in linear time [47]. However, we experimented with the DTW metric and we found it inadequate for ECG signals – the main problem is that this metric is by construction unable to preserve the position of the inner peaks in the compressed representations. Thus, in this work we resized each segment to a common length, as explained above, and checked for the best matching codeword using through a suitable distance function, as we explain next.

5) Codebook manager: this block has a key role in the proposed online compression scheme. It is loosely based on *vector quantization* [48] and has two main functions: 1) to *maintain* a consistent and representative codebook (dictionary) and 2) to *encode* input patterns into the corresponding indices from the codebook. Let $\boldsymbol{z}_t$ be the segment provided by the segment extraction block at the generic time $t = 0, 1, 2, \ldots$ (discrete time is assumed, corresponding to the arrival of a new segment). With $\mathcal{C}_t = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\}$ we indicate the codebook at time t, where $\boldsymbol{c}_i$, $i = 1, \ldots, N$, are the codewords therein. Segment $\boldsymbol{z}_t$ is remapped into a new segment $\boldsymbol{x}_t$ of length W samples as described above, where $\texttt{size}(\boldsymbol{c}_i) = \texttt{size}(\boldsymbol{x}_t) = W$, for $i = 1, \ldots, N$. The new segment $\boldsymbol{x}_t$ is obtained using linear resampling and removing offset o_t and gain g_t from $\boldsymbol{z}_t$ (see equations (5)–(7) of [43]). Thus, a suitable distance function

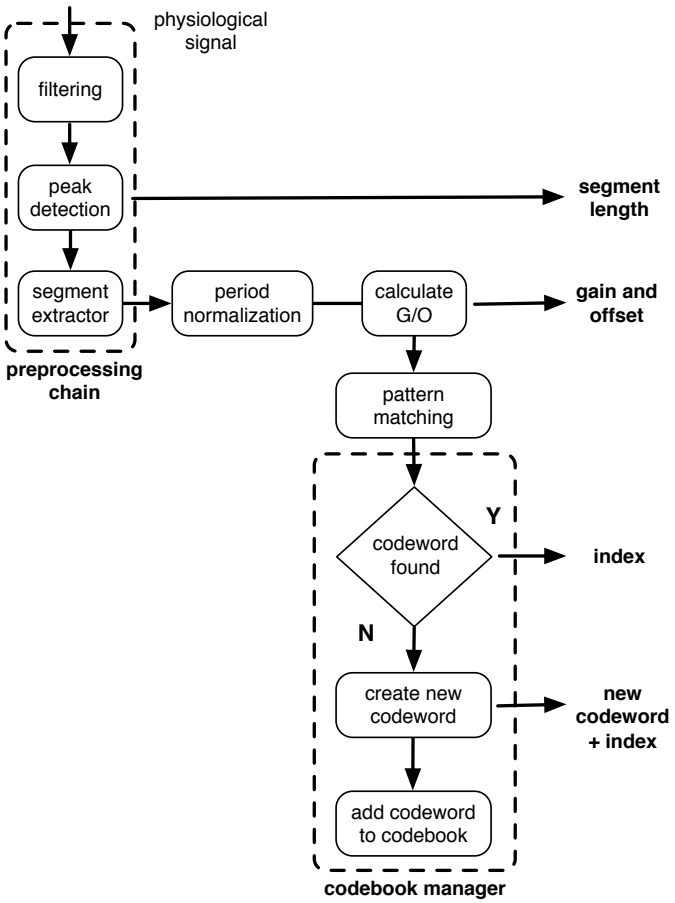

Fig. 1: Online codebook-based compression scheme.

$d(\boldsymbol{x}_t, \boldsymbol{c}_i)$ is evaluated for all codewords $\boldsymbol{c}_i$ in the codebook and the one with the minimum distance, with index $i^\star$, is picked. Now, if $d(\boldsymbol{x}_t, \boldsymbol{c}_{i^\star}) \leq \varepsilon$, codeword $\boldsymbol{c}_{i^\star}$ is deemed a good representative for the current segment $\boldsymbol{z}_t$, otherwise $\boldsymbol{x}_t$ is added to the codebook as a new codeword, where with $i^\star$ we mean the associated index. ε is a tunable parameter that we use to control the signal reconstruction fidelity at the decompressor. Finally, the index $i^\star$ is sent in place of the full segment, along with o_t, g_t and the original segment length, ℓ_t. The whole process is detailed in Fig. 1 (codebook manager block): if a match for $\boldsymbol{z}_t$ is found in the codebook (i.e., a codeword providing a sufficiently good accuracy, according to ε), then the corresponding index is sent over the transmission channel, along with the original segment length, its offset and gain parameters. These quantities correspond to the compressed bitstream, which is used at the decompressor to approximate the original time series by reversing each operation. Specifically, the decompressor applies three transforms to codeword $i^\star$ from the codebook: renormalization with respect to offset o_t and gain g_t and resampling according to the actual segment length ℓ_t. Otherwise, if no match is found for $\boldsymbol{z}_t$ at the compressor, this segment is added to the codebook as a new codeword and its normalized version (W samples) and the corresponding index are transmitted to the decoder so that the dictionary at the sender and that at the receiver remain synchronized at all

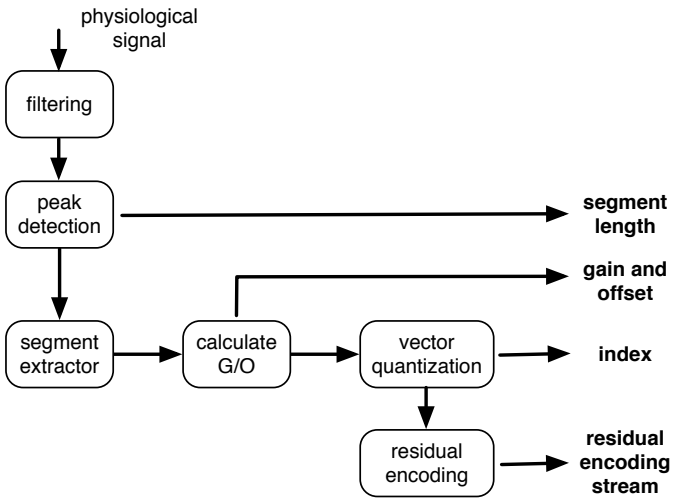

Fig. 2: Diagram of the GSVQ compression technique.

times.

We remark that several distance functions can be used in the codebook manager, the L_∞-norm has been considered for the results in this paper as it performed satisfactorily across a large range of signals.

According to our numerical results, as we show in Section IV, the number of codewords in the dictionary increases with decreasing ε but it tends to converge as time goes on. So the accuracy parameter ε also directly affects the dictionary size and, in turn, the memory requirements of the proposed algorithm. In case the codebook shall grow larger than the allowed memory space, the removal of codewords from the codebook can be implemented based on *last used* timestamps.

B. Gain-Shape Vector Quantization (GSVQ)

In this section we review the Gain-Shape Vector Quantization (GSVQ) method of [38]. The rationale behind this algorithm is to exploit the information redundancy among adjacent heartbeats by segmenting the ECG signal into segments and normalizing the period to a fixed length and amplitude. The normalized heartbeats are then used to build a dictionary having a fixed number of codewords K, through the Linde-Buzo-Gray algorithm [49]. While the general compression principle (i.e., inter-segment correlation) is similar to that in our online dictionary based scheme, GSVQ builds the codebook through an *offline* training phase.

Once the dictionary is obtained, the method associates each normalized heartbeat with the closest codeword, and sends the codeword index in place of the original time series. The algorithm also encodes the offset, the gain, and the length of the original segment, see Fig. 1. As a last step, the encoder calculates the residual, i.e., the difference between the current heartbeat (i.e., ECG segment) and the selected codeword, and uses the AREA algorithm [50], an adaptive sampling scheme for one dimensional signals, which obtains additional information to increase the quality of reconstruction. The principle behind the residual encoding phase is to encode and send a small number of significant points so as to bound the reconstruction error.

The decoder, upon receiving an encoded packet, retrieves the corresponding codeword from its local copy of the dictionary, performs a denormalization using the gain, the offset, and the length, and adds the residual stream to the reconstructed signal, see Fig. 2. As we shall see below, GSVQ performance predominantly depends on its residual encoding phase. The threshold used for residual encoding is in fact the main responsible for the amount of data to be transmitted, affecting the performance in terms of compression, reconstruction error, and energy efficiency.

C. Principal Component Analysis (PCA)

The goal of Principal Component Analysis (PCA) [13] is to shrink the information provided by a large set of correlated variables into a set of principal components with lower dimensionality. Each principal component is computed as a linear combination (linear transform) of the original variables, and the combination weights are chosen so that the components are mutually uncorrelated. This technique has been successfully applied in a multitude of applications, including ECG signal compression [34].

Before applying PCA, the biomedical signal goes through the preprocessing chain of Fig. 1, i.e., filtering, peak detection and segment extraction, where at time $t = 0, 1, 2, \ldots$ the last block normalizes each input segment $\boldsymbol{z}_t$ to a common length of W samples. The new segment is then stored into a vector $\boldsymbol{x}_t \in \mathbb{R}^W$ and is fed to the PCA encoder. Specifically, let $\boldsymbol{\mu_x} = E[\boldsymbol{x}^t]$ and $\boldsymbol{R_x} = E[\widetilde{\boldsymbol{x}}^t\widetilde{\boldsymbol{x}}^T_t]$ respectively be the mean of $\boldsymbol{x}_t$ and its covariance matrix, with $\widetilde{\boldsymbol{x}}_t = \boldsymbol{x}_t - \boldsymbol{\mu_x}$. PCA amounts to apply an orthonormal linear transformation $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_W]$ to $\widetilde{\boldsymbol{x}}_t$, so that the elements $w_1, \ldots, w_W$ of the principal component vector $\boldsymbol{w} = \boldsymbol{\Psi}^T\widetilde{\boldsymbol{x}}_t = \boldsymbol{\Psi}^T(\boldsymbol{x}_t - \boldsymbol{\mu_x})$ are mutually uncorrelated. It can be shown that the i-th principal component is obtained as $w_i = \boldsymbol{\psi}_i\widetilde{\boldsymbol{x}}_t$, where $\boldsymbol{\psi}_i$ is the eigenvector corresponding to the i-th largest eigenvalue of $\boldsymbol{R_x}$, for $i = 1, \ldots, W$. The set of eigenvectors corresponding to the W principal components is obtained solving $\boldsymbol{R_x\Psi} = \boldsymbol{\Psi\lambda}$ for $\boldsymbol{\Psi}$, where $\boldsymbol{\lambda}$ is a diagonal matrix containing the eigenvalues $\lambda_1, \ldots, \lambda_W$, placed in decreasing order. As the theoretical covariance matrix $\boldsymbol{R_x}$ is difficult to compute, a matrix $\boldsymbol{X} \in \mathbb{R}^{W\times m}$ is built by stacking m successive ECG segments: their sample mean $\hat{\boldsymbol{\mu}}_{\boldsymbol{x}}$ and their sample covariance matrix $\hat{\boldsymbol{R}}_{\boldsymbol{x}} = (\boldsymbol{X}\boldsymbol{X}^T)/m \in \mathbb{R}^{W\times W}$ respectively replace $\boldsymbol{\mu_x}$ and $\boldsymbol{R_x}$ for the calculation of the eigenvectors.

According to the above discussion, we can write $\boldsymbol{x}_t = \boldsymbol{\mu_x} + \boldsymbol{\Psi w}$ and, if the signal is sufficiently correlated, only a fraction of the weights in $\boldsymbol{w}$ suffices to accurately describe the input vector $\boldsymbol{x}_t$. Compression is thus achieved by applying the PCA transform and sending the desired number h of principal components, i.e., the first h weights in $\boldsymbol{w}$, with $h \leq W$. In Section III-D, we follow a similar rationale by using a particular neural network instance called autoencoder, which practically acts as a non-linear PCA [51].

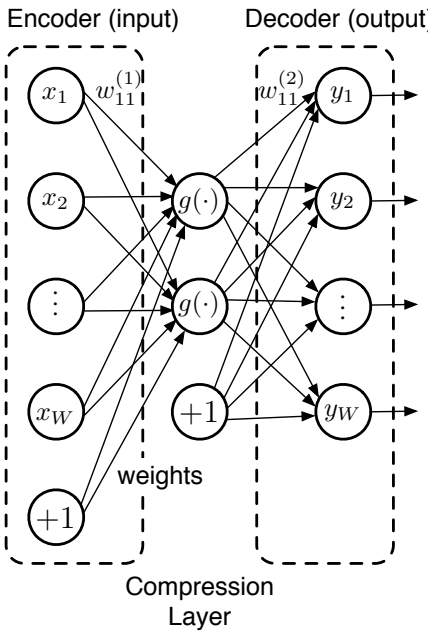

Fig. 3: Graphical representation of an autoencoder: input and output layers have the same dimension W, whereas the compression layer has $h = 2$ neurons. $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is assumed to be the logistic activation function $g(z) = (1 + \exp(-z))^{-1}$.

D. Autoencoders (AE)

An autoencoder [52] is a neural network where input and output layers have the same dimension W, whereas the deepest hidden layer has a smaller dimension h, with $h < W$, as we show in Fig. 3. With $w_{ij}^{(1)}$ ($w_{ij}^{(2)}$) we indicate the autoencoder weights from neuron i to neuron j of the input (output) layer. Here, autoencoders are used as a non-linear dimensionality reduction technique to compactly represent the information in the original segments (of size W) into a much smaller space (ideally $h \ll W$ neurons).

The training of this neural network is accomplished through an unsupervised learning algorithm that uses a number of training examples $\boldsymbol{x} \in \mathbb{R}^W$ that are placed at the input of the autoencoder. Specifically, backpropagation is executed by setting the output $\boldsymbol{y} = \boldsymbol{x}$ so that the neural network weights $w_{ij}^{(1)}, w_{ij}^{(2)}$ are adjusted for the autoencoder to behave as an identity function. In this work, we consider the approach of [18] where the authors use denoising autoencoders [53] to approximate the input biomedical patterns.

Once the autoencoder is trained to represent the input data, weights $w_{ij}^{(1)}$ fully specify the compressor (encoder), whereas $w_{ij}^{(2)}$ specify the decompressor (decoder), see Fig. 3. Signal compression is achieved by applying the preprocessing chain of Fig. 1, i.e., filtering, peak detection and segment extraction. Note that the last block also normalizes each segment to a common length of W samples. Each of such segments is inputted to the encoder section of the autoencoder, which returns the h values associated with the neurons in the compression layer. These h values correspond to the compressed representation of the current segment and are sent to the decompressor along with the original segment length. Finally, the decompressor at the receiver uses the values of these h inner neurons, along with weights $w_{ij}^{(2)}$, to obtain the reconstructed W-sample vector $\boldsymbol{y}$ through the decoder of Fig. 3. $\boldsymbol{y}$ is finally resized to the original segment length.

We remark that AE also belongs to the inter-segment correlation class of algorithms as it exploits the fact that patterns across different segments have a quasi-periodic behavior.

E. Compressive Sensing (CS)

Compressive sensing (CS) is a recently proposed theory [54] [55] to efficiently acquire and reconstruct a signal, by solving ill-posed linear systems of equations. This technique is based upon the premise that the signal of interest is sparse in some transform domain. This means that, the original signal can be represented in a domain where only a few transform coefficients are required for its full description. To be more specific, let $\boldsymbol{x} \in \mathbb{R}^W$ be an W-sized vector and assume that this vector can be represented in a K-sparse domain through the sparse vector $\boldsymbol{s}$, where only $K \ll W$ elements of $\boldsymbol{s}$ are non-zero, i.e., vector $\boldsymbol{s}$ is K-sparse in this domain. If we refer to the sparsification basis as $\boldsymbol{\Psi} \in \mathbb{R}^{W \times W}$, we have that $\boldsymbol{x} = \boldsymbol{\Psi s}$. Now, let $\boldsymbol{\Phi} \in \mathbb{R}^{m \times W}$ be a sampling matrix. Note that, using this matrix to sense the full signal $\boldsymbol{x}$, we have $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{n}$, where $\boldsymbol{n} \in \mathbb{R}^m$ represents the measurement noise, $\boldsymbol{y} \in \mathbb{R}^m$ and $m < W$, which means that $\boldsymbol{x}$ is being subsampled.

CS tools allow the recovery of $\boldsymbol{x}$ from its subsampled version $\boldsymbol{y}$, where: $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{n} = \boldsymbol{\Phi\Psi s} + \boldsymbol{n}$. This is achieved solving for $\boldsymbol{s}$ the following equation:

$$\min \|\boldsymbol{s}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{\Phi\Psi s}\|_2 \leq \epsilon \tag{1}$$

where ϵ represents a bound on the measurement noise. Numerically, a high number of techniques are available to solve (1); among them we cite ℓ_1-magic [56] subspace pursuit [57] and NESTA [58].

In this work, we consider two recent ECG compression algorithms from [16] and [59], which are based on CS. The former exploits a technique called Simultaneous Orthogonal Matching Pursuit (SOMP), whereas the latter uses Block Sparse Bayesian Learning (BSBL) [60]. The algorithms are introduced next.

1) SOMP-based CS compression technique (SOMP-CS): the **encoder** operates according to the following steps:

- **Peak detection:** similarly to codebook-based schemes, a peak detection method is applied to the input signal to decompose it into segments $\boldsymbol{x}_t$, $t = 0, 1, 2, \ldots$.
- **Period normalization:** each segment $\boldsymbol{x}_t$ is normalized to a common length (W samples) using cubic-spline interpolation. After that, the sparse representation is constructed using Daubechies wavelets (db4) [61].
- **Sampling and quantization:** each 6 consecutive ECG segments are stored into a $W \times 6$ matrix $\boldsymbol{X}$. A CS sampled matrix $\boldsymbol{Y}$ is then obtained as $\boldsymbol{Y} = \boldsymbol{\Phi X}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{m \times W}$ is a suitable sampling matrix, with $m \ll W$. $\boldsymbol{Y}$ and the corresponding original lengths are quantized and sent to the decoder. Note that this implementation of

CS belongs to both the inter- and the intra-segment class as matrix $\boldsymbol{Y}$ spans across different adjoining segments.

The **decoder** works as follows:

- **Simultaneous Orthogonal Matching Pursuit:** each segment is recovered from $\boldsymbol{Y}$ using the modified Simultaneous Orthogonal Matching Pursuit [62], which exploits the structure of the wavelet coefficients.
- **Period Recovery:** the reconstructed segments are re-interpolated according to their original lengths.

Note that SOMP-CS considers a number of subsequent segments (6 in the above description) and, in turn, also accounts for the "inter-segment" correlation structure of the ECG signal.

2) Block Sparse Bayesian Learning (BSBL): in CS frame-works the measured signal is written as $\boldsymbol{y} = \boldsymbol{\Psi}'\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{y} \in \mathbb{R}^m$ is the compressed vector, $\boldsymbol{\Psi}' = \boldsymbol{\Phi}\boldsymbol{\Psi}$, with $\boldsymbol{\Psi}' \in \mathbb{R}^{m \times W}$ is a suitable transformation matrix ($m \ll W$), $\boldsymbol{x} \in \mathbb{R}^W$ is a sparse vector (being in the transform domain) and $\boldsymbol{n} \in \mathbb{R}^m$ the noise vector. Generally, vector $\boldsymbol{x}$ has additional structure and can be further represented as a concatenation of a certain number g of blocks x_i, possibly having different length d_i so that $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_g)^T$. Each block $\boldsymbol{x}_i \in \mathbb{R}^{d_i}$, $i = 1, \ldots, g$, is assumed to satisfy a parametrized multivariate Gaussian distribution $p(\boldsymbol{x}_i, \gamma_i, \boldsymbol{B}_i) \sim \mathcal{N}(0, \gamma_i \boldsymbol{B}_i)$ with the unknown parameters γ_i and $\boldsymbol{B}_i$. $\gamma_i \geq 0$ controls the block-sparsity of $\boldsymbol{x}_i$ and when $\gamma_i = 0$ the i-th block becomes the all zero vector. $\boldsymbol{B}_i \in \mathbb{R}^{d_i \times d_i}$ is a positive definite matrix which captures the correlation structure within the i-th block. Assuming that the sub-blocks $\boldsymbol{x}_i$ are uncorrelated the prior of $\boldsymbol{x}$ is $p(\boldsymbol{x}, \{\gamma_i, \boldsymbol{B}_i\}) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_0)$, where $\boldsymbol{\Sigma}_0 = \mathrm{diag}\{\gamma_1 \boldsymbol{B}_1, \ldots, \gamma_g \boldsymbol{B}_g\}$. For the noise, it is assumed that $p(\boldsymbol{n}, \lambda) \sim \mathcal{N}(0, \lambda \boldsymbol{I})$, where $\lambda \in \mathbb{R}^+$ and $I \in \mathbb{R}^{m \times m}$ is the identity matrix. The posterior of $\boldsymbol{x}$ (given the measured vector $\boldsymbol{y}$) is thus obtained as

$$p(\boldsymbol{x}|\boldsymbol{y}; \{\gamma_i, \boldsymbol{B}_i\}_{i=1}^g) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}}) \tag{2}$$

where $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ can be readily dervied from λ, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Psi}'$. Finally, the Maximum-A-Posteriori (MAP) estimate of $\boldsymbol{x}$, denoted by $\hat{\boldsymbol{x}}$, is given by [60]:

$$\hat{\boldsymbol{x}} = \boldsymbol{\Sigma}_0 (\boldsymbol{\Psi}')^T \left[\lambda \boldsymbol{I} + \boldsymbol{\Psi}' \boldsymbol{\Sigma}_0 (\boldsymbol{\Psi}')^T\right]^{-1}. \tag{3}$$

Thus, the problem boils down to the estimation of the parameters λ and $\{\gamma_i, \boldsymbol{B}_i\}_{i=1}^g$. This is achieved using a Type II maximum likelihood procedure. Also, different techniques have been developed according to whether the block partition is known or not, see [60].

According to the BSBL algorithm, the ECG signal is split into a number of segments $\boldsymbol{x}$, each of which consists of W samples, where W is a tunable parameter. Typical values for m and W are $m = 256$ and $W = 512$, see [59]. The maximum compression efficiency is thus given by $W/m = 2$ (in Section IV, we experiment with different (m, W) pairs). We observe that BSBL accounts for the intra-block correlation without considering the correlation structure among subsequent ECG segments. We thus classify BSBL as an "intra-segment" compression scheme.

The implementations provided by the authors of [16] (SOMP-CS) and [59] (BSBL-CS) were used for the numerical results of Section IV.

F. Discrere Cosine Transform (DCT)

In the signal compression field, Discrete Cosine Transform (DCT) is often preferred to the Fourier Transform due to its superior energy compaction capabilities and the fact that it entails the use of real coefficients. Several ECG compression methods exploiting DCT have been proposed in the literature [63]–[68]. Basically, in all of the proposed algorithms DCT is used to reduce the amount of data to be sent through the transmission of a subset of transform coefficients, i.e., those which carry more information. Some solutions employ advanced techniques for the pre/post processing of the DCT coefficients that, however, for wearable devices are expected to be energetically prohibitive.

In this paper, we consider two DCT based compression methods that differ in the adopted coefficient selection approach:

- **DCT-Cardinality Thresholding:** with this selection method the number of coefficients to be retained is given as input, and the coefficients are added starting from the lowest frequencies, i.e., the leftmost coefficient. Through this strategy the compression ratio can be finely tuned, but there are no guarantees on the reconstruction error at the decompressor.
- **DCT-Energy Thresholding:** with this method the coefficients are selected so as to meet an energy threshold constraint. The total energy of the DCT spectrum, E, is calculated and the coefficients that contain a predetermined fraction E_{th} of this energy are kept. The coefficients are selected again from the lowest to the highest frequencies, exploiting the energy compaction property of the DCT, so that their frequency position does not have to be encoded.

G. Discrere Wavelet Transform (DWT)

Wavelet compression schemes are based upon the transmission of a subset of the transform coefficients. In fact, in the Wavelet domain most of the signal information is often concentrated in just a few of them. Letting $z[n]$ be the discrete input temporal signal, defined for $n = 0, 1, \ldots, M-1$, in this work we consider the Discrete Wavelet Transform (DWT) [61], which is defined through the following equations

$$\begin{aligned} &\gamma[j,k] = \frac{1}{\sqrt{M}} \sum_n z[n] \varphi_{j,k}[n]\,, \\ &\text{where} \quad \varphi_{j,k}[n] = \frac{1}{\sqrt{s_0^j}} \varphi\left[\frac{n - k\tau_0 s_0^j}{s_0^j}\right], \quad j,k \in \mathbb{N}\,. \end{aligned} \tag{4}$$

$\gamma[j,k]$ is the DWT coefficient matrix and $z[n]$ can be expressed as $z[n] = (1/\sqrt{M}) \sum_{j,k} \gamma[j,k] \varphi_{j,k}[n]$. The parameter s_0, called *scale step*, has a fixed value (greater than 1), and τ_0 is

the *translation step*. Dyadic sample is commonly used, which amounts to setting $s_0 = 2$ and $\tau_0 = 1$. $\varphi(\cdot)$ is a function, referred to as *mother wavelet*, that is translated and scaled to represent the original signal $z[n]$. An efficient and widely adopted implementation of DWT, both for decomposition and reconstruction uses Quadrature Mirror Filters (QMF). As shown in (4), $z[n]$ can be decomposed into an infinite number of wavelets, but in practice a few levels of decomposition (i.e., a finite number of Wavelet coefficients) already account for most of the signal energy. In the case of dyadic sampling, at each decomposition step the frequency band of the output signal is halved – this allows downsampling the output of each processing step by a factor of 2, without any loss of information.

Wavelet based compression uses the most significant coefficients for reconstruction and the coefficient selection strategy is the main discriminating factor among the existing algorithms. The most widely adopted selection strategies are:

- **DWT-Level Thresholding:** all the coefficients falling below a certain threshold are discarded. Usually, each level of decomposition has a different threshold. This method is commonly used for denoising.
- **DWT-Cardinality Thresholding:** this is the counterpart of "DCT-Cardinality Thresholding". Here, a fixed number of coefficients is retained, discarding those with the lowest absolute values. As for DCT, this selection strategy allows fine tuning the compression ratio, which directly depends on the number of coefficients retained. With this approach, it is however difficult to precisely control the resulting reconstruction quality [69].
- **DWT-Energy Thresholding:** this is the counterpart of "DCT-Energy Thresholding". An energy threshold $E_{\text{th}} < 1$ is set. At each step of the Wavelet transform, the coefficient with the highest value is retained (i.e., it is included in the compressed vector $\boldsymbol{y}$). The energy of the selected coefficients is defined as $E = \boldsymbol{y}^T\boldsymbol{y}$. This operation is repeated until $E/E_0 > E_{\text{th}}$, where E_0 is the energy of the input signal $z[n]$.

In this work, we implemented the algorithm of [15], which is based on energy thresholding. There, in addition to the vector containing the retained coefficients, a coefficient map is also sent so as to track their position within the considered transform levels. The compression ratio is then tuned via E_{th}. Five levels of dyadic decomposition were considered, and bi-orthogonal 4.4 was used as the mother wavelet, as it provided the best results among other choices.

H. Lightweight Temporal Compression (LTC)

LTC [12] is a fast linear approximation technique working as follows. Let $z[n]$, $n = 0, 1, 2, \ldots$ be the input time series. The algorithm starts selecting $z[0]$ as the left endpoint of the current approximating segment. The following points $z[n]$ with $n > 0$ are transformed into vertical intervals $[z[n]-\varepsilon, z[n]+\varepsilon]$ where $\varepsilon > 0$ is an error tolerance on the reconstructed signal. When point $n > 1$ is considered, LTC evaluates the segment with extremes $(z[0], z[n])$ and checks whether this segments falls within *each* of the previously obtained vertical intervals around $z[1], z[2], \ldots, z[n-1]$. If this is the case, the algorithm obtains the vertical interval for the current point n and performs the check for the next point $n+1$. Otherwise, the algorithm stops, taking $z[n-1]$ as the right endpoint of the current segment. Thus, 1) $z[0]$ and $z[n-1]$ are sent as the left and right endpoints of the current segment as an approximation to values $\{z[0], z[1], \ldots, z[n-2], z[n-1]\}$ and 2) the algorithm reiterates with a new approximating segment, taking $z[n-1]$ as its left endpoint.

IV. Numerical Results

In this section, we show quantitative results for the considered signal compression algorithms, detailing their energy consumption (for compression and transmission), compression efficiency and reconstruction fidelity. The energy consumption for compression has been computed by taking into account the number of operations performed by the Micro-Controller Unit (MCU), i.e., the number of summations, multiplications, divisions and comparisons. These have been subsequently translated into the corresponding number of cycles and, in turn, into the energy consumption in Joule per bit considering the Cortex M4 [70] as a reference processor, see also [71]. In addition to the processing energy, we also considered the energy consumption associated with the transmission of the compressed data over the wireless medium. To this end, we took a Texas Instruments CC2541 low-energy Bluetooth system-on-chip [72], which is widely adopted for IoT devices.

The **Compression Efficiency (CE)** has been computed as the ratio between the total number of bits that would be required to transmit the full signal divided by those required for the transmission of the compressed bitstream. For the reconstruction fidelity, we computed the **Root Mean Square Error (RMSE)** between the original and the compressed signals normalizing it with respect to the signal's peak-to-peak amplitude, that is:

$$\text{RMSE} = \frac{100}{\text{p2p}} \sqrt{\frac{\sum_{i=1}^{L}(x_i - \hat{x}_i)^2}{L}}\,, \tag{5}$$

where L corresponds to the total number of samples in the trace, x_i and $\hat{x}_i$ are the original sample and the one reconstructed after the decompressor in position i, respectively. p2p is the average peak-to-peak signal's amplitude.

We observe that other metrics such as the Percentage Root mean square Difference (PRD) are also possible [73]. Nevertheless, we observe that PRD does not have a direct interpretation, whereas our RMSE metric allows one to immediately gauge the error against the signal's range. For this reason, we used (5) in the following plots.

In the next Section IV-A we first assess the performance of the considered compression algorithms for the standard test ECG traces from the PhysioNet MIT-BIH arrhythmia database [74]. Thus, in Section IV-B we extend our analysis to ECG traces that we collected from a Zephyr BioHarness 3 wearable chest strap. In Section IV-C, we consider PPG and RESP signals.

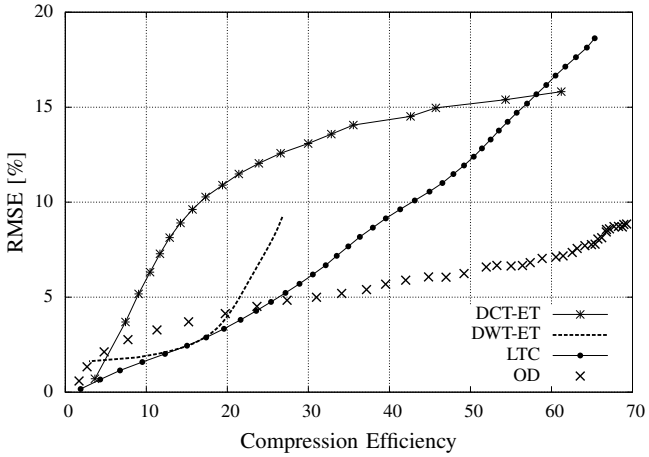

Fig. 4: RMSE *vs* compression efficiency for ECG signals: DCT, DWT, LTC and OD.

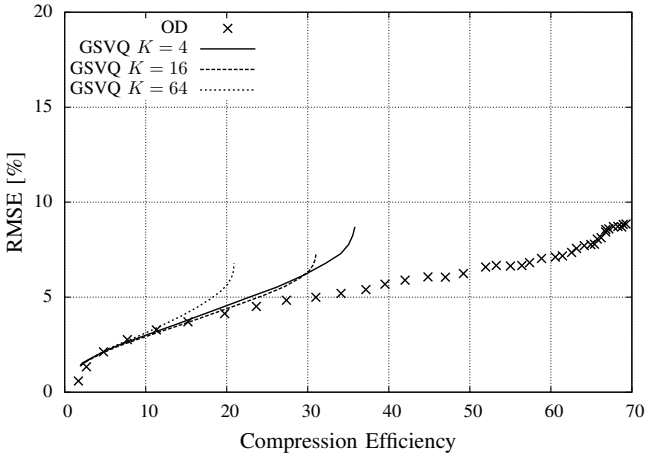

Fig. 5: RMSE *vs* compression efficiency for ECG signals – comparison of codebook-based compression schemes: GSVQ and OD. K is the (fixed) size of the GSVQ dictionary.

A. PhysioNet ECG traces

In the first set of graphs, we show results for ECG signals. To this end, we considered the following traces from the MIT-BIH arrhythmia database [74]: 101, 112, 115, 117, 118, 201, 209, 212, 213, 219, 228, 231 and 232, which were sampled at rate of 360 samples/s with $11-$bit resolution. Note that not all the traces of the database are usable (some are very noisy due to heavy artifacts probably due to the disconnection of the sensing devices) and an educated selection has to be carried out for a meaningful performance analysis, as done in previous work [74], [75]. The above performance metrics were obtained for these traces and their average values are shown in the following plots.

In Figs. 4, 5 and 6 we show the RMSE *vs* CE performance for all compression algorithms. Fig. 4 shows the performance of standard compression approaches, namely, DCT, DWT and linear approximation (LTC), Fig. 5 presents that of the codebook-based schemes (GSVQ and OD), whereas in Fig. 6 we show results for dimensionality reduction algorithms, namely, BSBL-CS, SOMP-CS, PCA and AE. Our online dictionary (OD) scheme is plotted in all figures for comparison. The tradeoff curves of OD have been obtained by varying the representation accuracy ε as a free parameter.

In Fig. 4, we consider the energy thresholding version of DCT (DCT-ET). We also experimented with its cardinality thresholding (CT) variant and we found its performance to be very similar to that of DCT-ET in every respect (RMSE, compression efficiency and energy). Thus, implementation convenience will dictate which of the two variants is to be preferred. LTC outperforms DCT-ET in terms of RMSE and CE; although, we remark that this is not always the case. For example, in [36] a DCT implementation that considerably surpasses LTC in terms of RMSE is proposed, but this comes at the price of a much higher computational complexity. This is possible through a more sophisticated selection of the coefficients, which requires performing inverse transforms for every ECG segment. This DCT variant is however not considered here as it is not deemed appropriate for wearable

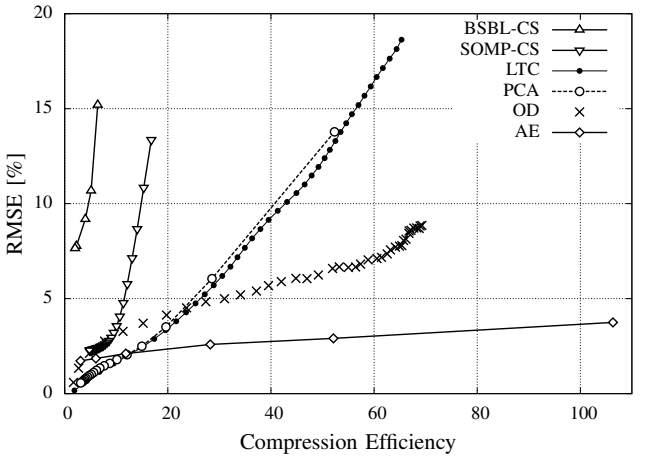

Fig. 6: RMSE *vs* compression efficiency for ECG signals: BSBL-CS, SOMP-CS, PCA, LTC, AE and OD.

devices, due to its high computational cost. DWT does a much better job than DCT in terms of RMSE, especially at relatively small compression efficiencies, say, smaller than 30, but it is unable to reach higher compression efficiencies, for which LTC and OD are to be preferred. At small compression efficiencies, adaptive algorithms may be a valuable option – for instance, one may switch between LTC and OD as a function of the required compression level. For OD, we also look at the number of codewords in the dictionary as a function of the compression efficiency, see Fig. 7. From that plot, we see that using OD is especially convenient at high compression efficiencies, i.e., higher than 45. In this region, the size of the dictionary is in fact reasonably small (smaller than 35 codewords) and it is thus feasible to store it in the limited memory of wearables. Specifically, for the considered setup the size of each codeword is $(W \times 11)/8 = 275$ bytes, where $W = 200$ is the segment length and 11 is the number of bits to represent the signal samples from the MIT-BIH database. This means that a dictionary of 35 codewords takes

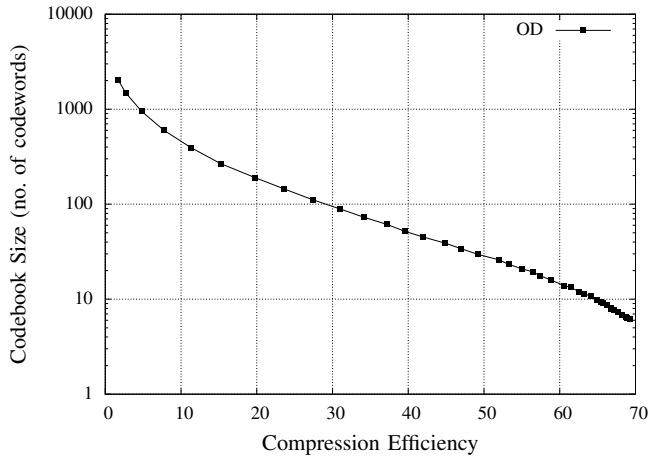

Fig. 7: Online dictionary compression: codebook size as a function of the compression efficiency. The tradeoff curve is obtained by varying the representation accuracy parameter ε.

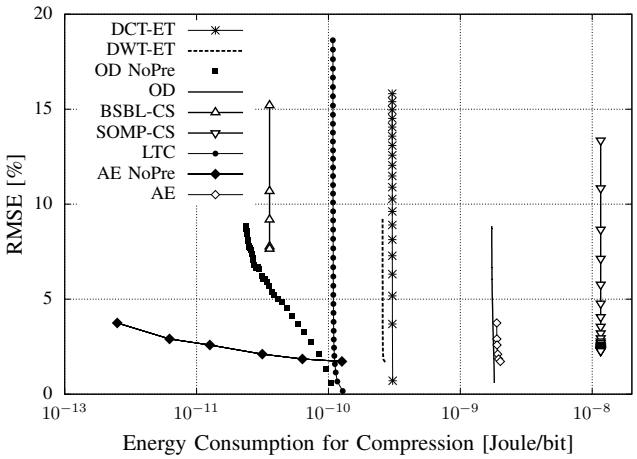

Fig. 8: RMSE as a function of the energy consumption associated with the compression of ECG signals.

$35 \times 275 = 9.625$ kbytes of memory space.

A comparison for the codebook-based algorithms is presented in Fig. 5. For GSVQ we move along the RMSE *vs* CE curves by changing the threshold that governs the number of bits that are encoded into the residual stream. As discussed in Section III-B, residual encoding is the operation that affects the most the performance of GSVQ. The dictionary size K affects the maximum achievable compression but the maximum CE is always smaller than that of OD, where the dictionary adapts to the signal in an online fashion. Although not shown in the plot, one may be thinking of not sending the residual encoding stream, so as to reach higher compression efficiencies. However, due to the use of a precomputed and fixed dictionary, this leads to a very high RMSE and is not recommended.

PCA is shown in Fig. 6. From this graph we see that the performance of PCA closely matches that of LTC, which is plotted in the same figure for the sake of comparison. This is quite interesting and non trivial – although both algorithms rely on linear approximations, PCA is rather involved, whereas LTC has a much lighter computational cost, as we show shortly. Also, in Fig. 6 the tradeoff curve for PCA is obtained by varying the number of principal components h from 100 (leftmost point in the figure) down to 5 (rightmost point) in steps of 5, whereas the performance of LTC is plotted varying ε within a continuous interval. Overall, LTC permits a fine-grained control of the RMSE *vs* compression tradeoff, whereas this is not possible with PCA, especially at high compression efficiencies (small h). Finally, LTC provides a means to precisely control the maximum reconstruction error, through the parameter ε, whereas the number of retained principal components h does not provide any guarantee in terms of reconstruction accuracy.[1]

AE is shown in Fig. 6, where the number of inner neurons h is varied as a free parameter in $\{100, 50, 25, 10, 5, 2\}$: $h = 100$ is represented by the leftmost point in the graph, whereas the rightmost corresponds to $h = 2$. AE obtains the best performance both in terms of RMSE and CE. We underline that this algorithm entails an *offline* training phase which has two drawbacks: 1) usually, this phase is computationally demanding and requires a representative dataset, 2) although autoencoders have excellent generalization capabilities, if the statistics of the underlying signal changes substantially, there are no guarantees that autoencoders trained with the old data will still provide good approximations for the new signals. However, the RMSE performance achieved by AE is striking and spurs the use of neural networks within this domain. A note on the AE compression efficiency is in order. For OD the compressed bitstream comprises the following fields, which are sent for each new ECG segment: the codeword index, the original segment length, the gain and the offset. Thus, when no updates of the dictionary occur, 4 parameters are to be sent for each new segment. The maximum compression efficiency of OD is thus obtained as[2] $CE_{\mathrm{OD}}^{\max} = \texttt{SamplesPerSeg}/4$, where `SamplesPerSeg` corresponds to the number of samples in a segment. With AE, for each segment we only send the h values associated with the inner neurons, see Fig. 3, and the length of the original segment. Two additional offsets are sent only once, when the compression starts. Thus, the maximum CE is approximated as $CE_{\mathrm{AE}}^{\max} \simeq \texttt{SamplesPerSeg}/(h+1)$. For an average segment size of 318 samples, we get $CE_{\mathrm{OD}}^{\max} = 80$ and $CE_{\mathrm{AE}}^{\max} = 106$, which explain the results in Fig. 3. Note that the compression efficiency of OD is actually smaller than 80 and this is due to the new codewords that must be sent to update the dictionary at the decompressor.

For AE, the memory occupation is fixed and amounts to the memory needed to store the weights of the encoder in Fig. 3 and the output values generated by the inner layer, i.e., $((W+1) \times h \times 11)/8$ bytes, where $W = 200$, h is the number of inner neurons and 11 is the number of bits

[1]With PCA, an inverse transform at the compressor is required to assess the reconstruction error provided by a certain value of h.

[2]For the sake of clarity, we assume that input samples and parameters are represented through the same number of bits. If this is not the case, the following equation should consider the different precision.

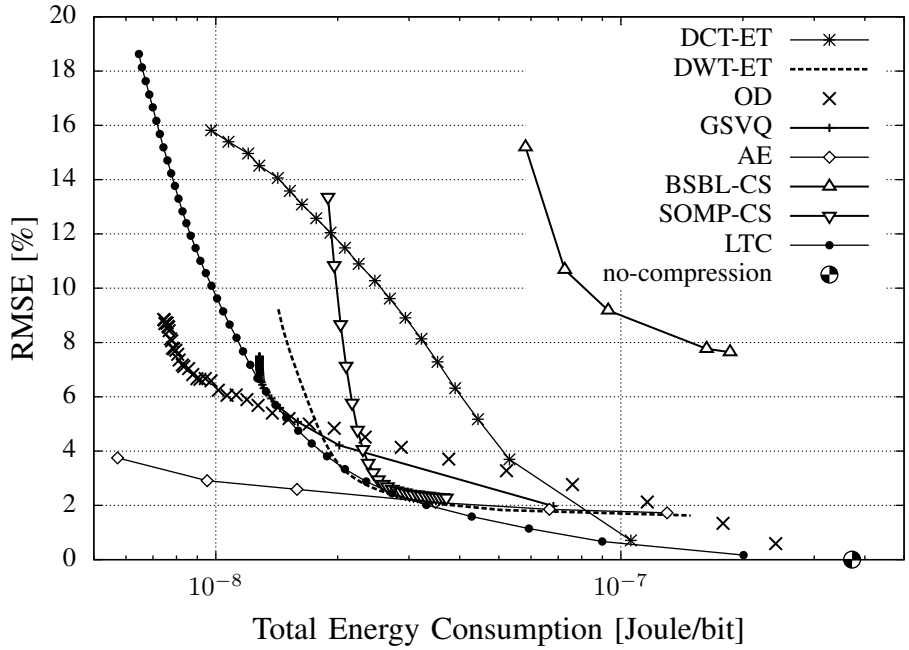

Fig. 9: RMSE as a function of the total energy consumption (compression plus transmission) of ECG signals.

TABLE I: Complexity [no. operations] and energy consumption [μJ] figures. RMSE is 7.5% for all algorithms except AE for which the average RMSE is 3.75% (the highest with AE, obtained with $h = 2$ neurons in the inner compression layer).

	DCT-ET	**DWT-ET**	**GSVQ**	**BSBL-CS**	**SOMP-CS**	**LTC**	**OD**	**AE**
Additions	13463	7809.9	98114	3747	540912	1258.8	84411	82439
Multiplications	9089.8	6883.5	82788	0	613797	0	82817	81535.7
Divisions	0	323.16	1137.2	0	311.89	634.05	940.19	938.689
Comparisons	30.35	7723.1	475.63	0	155.72	1231.1	986.99	462.708
Compression energy	0.74	0.88	6.47	**0.12**	38.05	0.37	5.95	5.83
TX energy	124.22	45.85	35.82	639.03	32.15	37.90	20.37	**13.45**
Total energy	124.96	46.73	42.29	639.15	64.20	38.27	26.32	**19.28**

to represent a floating point (either a weight or an encoder output). Given this, with, e.g., $h = 8$ the memory footprint of AE is 2.211 kbytes.

As for the CS-based algorithms, neither SOMP-CS nor BSBL-CS provides satisfactory performance. The compression efficiency of SOMP-CS is rather small and the corresponding RMSE tends to diverge for, e.g., CE larger than 12. As we shall see shortly below, the energy performance of SOMP-CS is unsatisfactory when compared to that of other algorithms and the compression strategy of BSBL-CS has the lowest energy consumption, but its intra-segment approach is much less effective than that of other inter-segment schemes such as SOMP-CS, dictionary based algorithms (GSVQ, OD) and AE. Although the results that we show here for SOMP-CS and BSBL-CS were respectively obtained using the implementations from [16] and [59], we found similar CE figures in other papers [17], where the compression efficiency is defined as $\text{CE}' = ((W - m)/W) \times 100$, with W being the number of original samples and m the number of compressed samples that are transmitted to the receiver. With this definition, CS schemes achieve maximum efficiencies of 80-90%. We observe that these figures correspond to a CE ranging from 5 to 10 according to the definition that we use in the present paper, i.e., $\text{CE} = W/m$.

In Fig. 8, we show the RMSE against the energy drained for compression at the transmitter, expressed in Joule per bit in the original ECG sequence. These tradeoff curves are obtained by varying the compression efficiency of each algorithm from 1 to the maximum achievable (which is scheme-specific, see Fig. 10). SOMP-CS has the highest energy consumption, BSBL-CS the smallest, LTC is the second best, whereas OD and AE perform between LTC and SOMP-CS. The good performance of BSBL-CS is due to its compression algorithm which just multiplies the input signal by sparse binary matrices, with entries in $\{0, 1\}$. We underline that the energy consumption of OD and AE is dominated by the preprocessing chain of Fig. 1. To verify this, in this figure we also show their performance by removing the contribution of the pre-processing blocks: the corresponding curves are referred to in the plot as "OD NoPre" and "AE NoPre", respectively. Note that filtering is always performed to remove measurement artifacts and peak detection is also very often utilized to extract relevant signal features. Given this, the energy consumption associated with the required pre-processing functions may not be a problem, especially if these functions are to be executed anyway. Although not shown for the sake of readability of the plot, PCA and GSVQ have nearly the same energy consumption of OD.

In Fig. 9, we show the RMSE as a function of the *total energy consumption*, which is obtained summing the energy required for compression to that for the subsequent transmission of the compressed bitstream over a CC2541 Bluetooth low-energy wireless interface. This total energy is then normalized with respect to the number of bits in the original ECG signal. From this plot, we see that the total energy consumption is dominated by the transmission energy, which depends on the compression efficiency. The best algorithms are LTC, OD and AE and the algorithm of choice depends on the target RMSE that, in turn, directly descends from the selected compression efficiency. As discussed above, an adaptive algorithm may be a good option, where for each value of CE the scheme providing the smallest RMSE is dynamically selected. The energy consumption when no compression is applied is also shown in the figure for comparison. We see that signal compression, and the subsequent reduction in size of the data to be transmitted, allows a considerable reduction in the total energy consumption. Specifically, AE and OD respectively enable energy savings of about one order of magnitude while respectively providing RMSEs smaller than 2% and 4%. The performance of AE is particularly striking as it allows saving up to two order of magnitude in terms of energy consumption by still keeping the RMSE around 4%. This motivates the use of signal compression techniques for continuous monitoring applications for IoT devices. Note that the actual RMSE can be dynamically tuned at runtime, by allowing slightly less accurate representations (and thus much higher compressions) when no critical patterns are detected. Also, for AE a visual inspection reveals that a RMSE smaller than 4% entails ex-cellent approximations to the original biosignals, and that the error is mainly due to smoothing out spurious oscillations that are introduced and that are not filtered by the preprocessing chain of Fig. 1.

A breakdown of the complexity and energy consumption figures for the considered algorithms is provided in Table I. These metrics were obtained for the Physionet ECG signals and represent the average complexity (expressed in terms of number of operations) and energy consumption (Joules) for the compression and transmission of a single ECG segment of data. As expected, BSBL-CS is the most energy efficient for the compression phase, whereas LTC is the second best. Other algorithms such as OD and AE are more demanding in terms of number of operations but their compression efficiency is much higher. Since the transmission energy dominates that needed for data processing, AE and OD represent the best alternatives when all the contributions are added up.

B. Wearable ECG Signals

We now present some results for ECG signals that we acquired from a Zephyr BioHarness 3 wearable device [6]. To this end, we collected ECG traces from eleven healthy individuals, which were continuously recorded during working hours, i.e., from 8am to 6pm. These were sampled at a rate of 250 samples/s with each sample taking 12 bits.

The RMSE *vs* CE tradeoff for these signals is shown in Fig. 10 for the best performing compression algorithms.

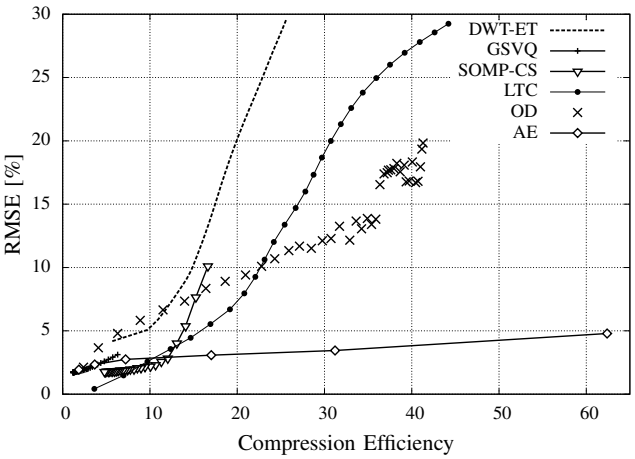

Fig. 10: RMSE *vs* compression efficiency for ECG signals: DWT-ET, GSVQ, SOMP-CS, LTC, OD and AE.

The results are similar to those of Figs. 4–6 with the main difference that in this case the ECG signals have more artifacts and a higher variability. As such, the resulting RMSE is also higher for all schemes and the compression performance is degraded. The general trends and recommendations remain unchanged, i.e., SOMP-CS and LTC are good choices at low up to intermediate compression efficiencies, whereas OD and AE perform better at higher CEs. However, we remark that the OD's compression efficiency is impacted with respect to that in Fig. 6 as the non-steady data of wearables requires more frequent dictionary updates. AE is still very effective, providing the highest compression efficiencies and the smallest RMSE. BSBL-CS performs unsatisfactorily and its RMSE performance is the one being affected the most by the non-steady behavior of wearable data.

The energy consumption figures of all schemes, although slightly rescaled, have a totally similar behavior as those obtained with the PhysioNet MIT-BIH traces (see Figs. 8 and 9) and are thus not shown in the interest of space. In fact, the energy consumption is marginally affected by the non-stationary behavior of wearable signals, which impacts more on the RMSE and compression performance.

As an illustrative example, in Figs. 11 and 12 we respectively look at the per segment RMSE and CE performance of LTC, OD and AE. In Fig. 11 we fix the compression efficiency to CE=28 for all schemes and we show the RMSE for each segment considering 12 minutes of ECG readings from one of the subjects. Overall, OD performs satisfactorily, providing an average RMSE of 4%, LTC settles around an RMSE of 11% and AE achieves the best accuracy, i.e., RMSE=2.6%. Artifacts and the non-steady behavior of the Bioharness ECG traces require more frequent dictionary updates for OD, which then entail some major variability in its RMSE performance, as can be clearly seen in the range $[600, 700]$ segments in both plots. Specifically, when the current dictionary is no longer representative of the input data, at first the RMSE increases and then it sharply decreases due to the consequent dictionary update. Fig. 12 shows the per segment compression

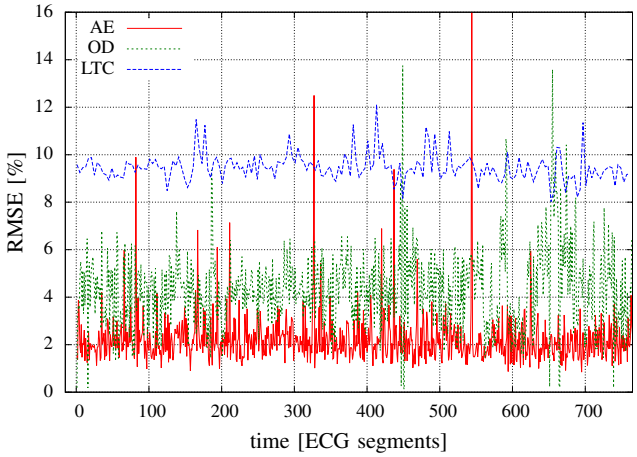

Fig. 11: RMSE as a function of time. CE=28 for both schemes, RMSE(LTC)= 11%, RMSE(OD)= 4% and RMSE(AE)= 2.6%.

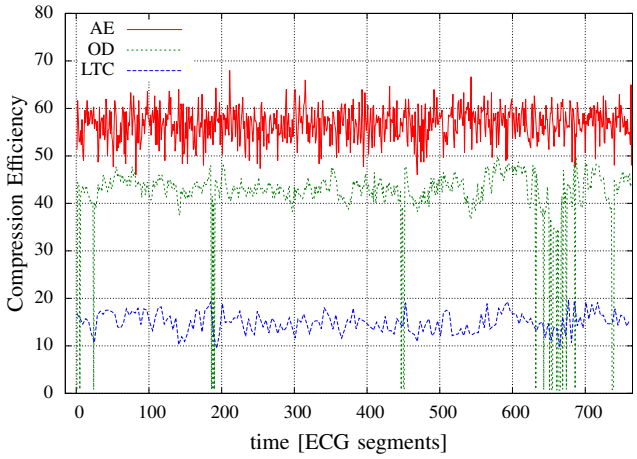

Fig. 12: CE as a function of time. RMSE = 3% for all schemes, CE(LTC)= 15, CE(OD)= 19 and CE(AE)= 56.

efficiency for the same ECG trace by operating LTC, OD and AE so that their average RMSE is 3%. From this plot we see that AE reaches much higher CEs, delivering strikingly good performance. Besides, the CE of OD sometimes drops to one (no compression) and this happens when the dictionary is updated (see the range $[600, 700]$ segments in the x-axis of Fig. 12). Note that the first update occurs at time zero, as OD has no dictionary at the beginning of the ECG sequence.

C. PPG and RESP Signals

As a final result, in Figs. 13 and 14 we respectively show the RMSE *vs* CE performance for PPG and respiratory (RESP) signals from the PhysioNet MIMIC-II waveform database [74]. In these graphs, we only show the performance of the three best algorithms, namely, LTC, OD and AE. AE is plotted considering the number of inner neurons $h \in \{100, 50, 25, 10, 5, 2\}$, where $h = 100$ is represented by the leftmost point, whereas $h = 2$ by the rightmost, and outperforms all the remaining schemes for $h \leq 5$. Clearly, OD

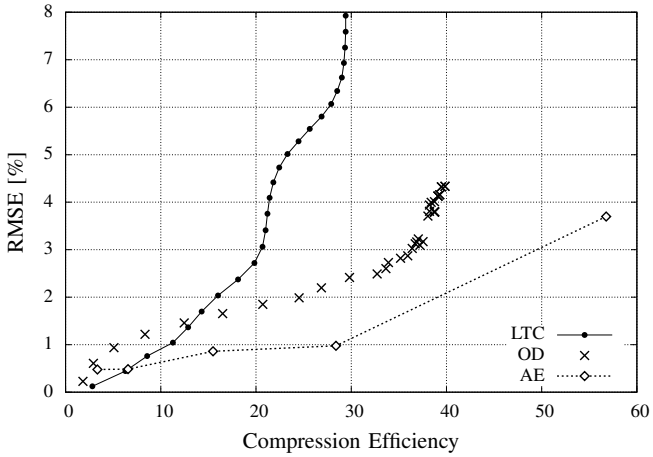

Fig. 13: RMSE *vs* compression efficiency for PPG signals.

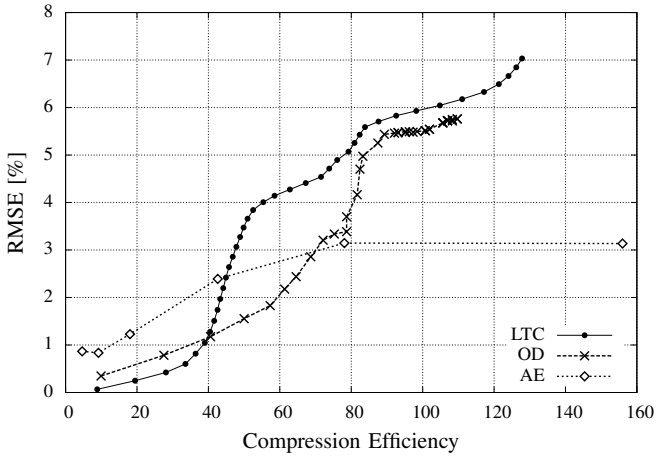

Fig. 14: RMSE *vs* compression efficiency for respiratory signals.

and AE are still effective for these signal types. For respiratory signals, LTC performs best for compression efficiencies up to 40, OD is to be preferred for intermediate efficiencies between 40 and 80. The compression efficiency obtained for PPG signals is smaller than that achieved for ECG and RESP but this is due to the lower sampling rate in the PPG traces. For all signals, the RMSE of AE never exceeds 4%, while its CE respectively reaches 56 and 156 for PPG ad RESP when just two inner neurons ($h = 2$) are utilized. These results are impressive and motivate further research, especially to make the AE learning phase online and subject-adaptive, as we do for OD.

V. Lesson Learned and Open Issues

In this paper, we advocated the use of lossy compression as a means to boost energy efficiency in wearable wireless devices. As a first contribution, we presented an original dictionary based technique, where compression is achieved by building and maintaining at runtime a dictionary. This dictionary is subsequently used to approximate signal sequences transmitting codeword indices in place of the original samples.

This technique is found to be very effective, showing excellent approximation capabilities and very high compression efficiencies at the cost of a reasonably small amount of computation. We then considered compression algorithms based on linear approximations. Despite their inherent simplicity, we found them to be quite effective and, when the required compression efficiency is not too high, they represent the best option among competing solutions. We also found that a recent scheme belonging to this class, called lightweight temporal compression, very closely matches the performance of principal component analysis, at a much smaller computational cost and additionally providing inbuilt guarantees on the maximum approximation error at the decompressor. Thus, we explored the performance of recent approaches based on autoencoders. These neural network architectures are found to be extremely effective, leading to the highest compression efficiencies at a reasonable computational cost. Their performance is striking especially at very high compression rates, where just two inner neurons are utilized to represent input patterns comprising several hundreds of points, still providing very small approximation errors (usually the RMSE remains bounded within 4%). The performance of these algorithms was numerically evaluated against that of the most prominent schemes from the literature, i.e., Fourier and Wavelet transforms, compressive sensing and vector quantization techniques.

Open research areas. From the numerical analysis that we have carried out in this paper, we have identified several avenues for future research. We have seen that the most promising means to reach high compression efficiencies is to exploit inter-segment correlation. Dictionary based algorithms belong to this category and do a very good job in all respects. Nevertheless, the online scheme proposed in this paper uses too much memory space at relatively small compression efficiencies, say, smaller than 40. Autoencoders also have a main drawback. In fact, these networks need to go through an offline training phase, during which their weights are shaped utilizing a representative dataset. Although they have excellent generalization capabilities, they will be nevertheless unable to satisfactorily represent input patterns that sharply differ from those in the dataset used for training. Hence, a desirable contribution would be to concoct a new neural network based algorithm with the following properties, both very relevant from a practical standpoint: a) we would like the size of the dictionary not to grow with a diminishing RMSE or, equivalently, with a decreasing compression efficiency. Ideally, the dictionary size should be kept constant. b) We would also like the training phase to be carried out in an online fashion. In this way, the dictionary will adapt to the specific signal statistics of the subject wearing the device. Another interesting subject for future investigations is the joint compression of multiple vitals, including respiratory rate, electrocardiogram, plethysmograph, and data from motion sensors.

ACKNOWLEDGMENT

The work in this paper has been supported by Samsung Advanced Institute of Technology (SAIT), Korea, as part of the SAMSUNG Global Research Outreach (GRO) program.

REFERENCES

[1] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric Sensing," *Phylosophical Transactions of Royal Society*, vol. 370, no. 1958, pp. 176–197, Jan. 2012.

[2] S. Riazul Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015.

[3] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A Review of Wearable Sensors and Systems with Application in Rehabilitation," *Journal of Neuroengineering and Rehabilitation*, vol. 9, no. 21, pp. 1–13, Apr. 2012.

[4] R. Istepanian, S. Hu, N. Philip, and A. Sungoor, "The potential of Internet of m-health Things "m-IoT" for non-invasive glucose level sensing," in *IEEE International Conference of the Engineering in Medicine and Biology Society (EMBC)* , Boston, MA, US, Aug. 2011, pp. 5264–5266.

[5] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *MDPI Sensors*, vol. 14, no. 6, pp. 10 146–10 176, Jun. 2014.

[6] Zephyr Technology Corporation, "Bioharness 3 - Wireless Professional Heart Rate Monitor and Physiological Monitor," 2015. [Online]. Available: http://www.zephyranywhere.com/

[7] M. Hesse, P. Christ, T. Hormann, and U. Ruckert, "A Respiration Sensor for a Chest-Strap Based Wireless Body Sensor," in *Proceedings of IEEE Sensors*, Valencia, Spain, Nov. 2014, pp. 490–493.

[8] Thought Technology Ltd, "Respiration Sensor: Model SA9311M," 2015. [Online]. Available: http://www.thoughttechnology.com/

[9] Seraphim Sense Ltd., "Angel open sensor wristband," 2015. [Online]. Available: http://www.angelsensor.com/

[10] Z. Zhang, Z. Pi, and B. Liu, "TROIKA: A General Framework for Heart Rate Monitoring Using Wrist-Type Photoplethysmographic Signals During Intensive Physical Exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522–531, Feb. 2015.

[11] R. Mukkamala, J.-O. Hahn, O. T. Inan, L. K. Mestha, C.-S. Kim, H. Toreyin, and S. Kyal, "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 1879–1901, Aug. 2015.

[12] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight Temporal Compression of Microclimate Datasets," in *Proceedings of the IEEE International Conference on Local Computer Networks (LCN)*, Tampa, FL, US, Nov. 2004, pp. 516–524.

[13] C. R. Rao, "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhyā: The Indian Journal of Statistics*, vol. 26, no. 4, pp. 329–358, Dec. 1964.

[14] R. Shankara and I. S. N. Murthy, "ECG Data Compression Using Fourier Descriptors," *IEEE Transactions on Biomedical Engineering*, vol. 33, no. 4, pp. 428–434, Apr. 1986.

[15] B. A. Rajoub, "An Efficient Coding Algorithm for the Compression of ECG Signals Using the Wavelet Transform," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 4, pp. 355–362, Apr. 2002.

[16] L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner, "Compressed Sensing Based Method for ECG Compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 761–764.

[17] G. D. Poian, R. Bernardini, and R. Rinaldo, "Gaussian Dictionary for Compressive Sensing of the ECG Signal," in *Proceedings of the IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS)*, Rome, Italy, Oct. 2014, pp. 80–85.

[18] D. Del-Testa and M. Rossi, "Lightweight Lossy Compression of Biometric Patterns via Denoising Autoencoders," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2304–2308, Dec. 2015.

[19] R. Francescon, M. Hooshmand, M. Gadaleta, E. Grisan, S. K. Yoon, and M. Rossi, "Toward Lightweight Biometric Signal Processing for Wearable Devices," in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, Milan, Italy, Aug. 2015.

[20] A. K. Jain, R. Chellappa, S. C. Draper, N. Memon, P. J. Phillips, and A. Vetro, "Signal Processing for Biometric Systems," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 146–152, Nov. 2007.

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, Dec. 2010.

[22] M. Elgendi, B. Eskofier, S. Dokos, and D. Abbott, "Revisiting QRS Detection Methodologies for Portable, Wearable, Battery-Operated, and Wireless ECG Systems," *PLoS ONE*, vol. 9, no. 1, p. e84018, Jan. 2014.

[23] P. Morizet-Mahoudeaux, C. Moreau, D. Moreau, and J. Quarante, "Simple microprocessor-based system for on-line e.c.g. arrhythmia analysis," *Medical and Biological Engineering and Computing*, vol. 19, no. 4, pp. 497–500, Jul. 1981.

[24] M. L. Ahlstrom and W. J. Tompkins, "Automated High-Speed Analysis of Holter Tapes with Microcomputers," *IEEE Transactions on Biomedical Engineering*, vol. 30, no. 10, pp. 651–657, Oct. 1983.

[25] Y. Chen and H. Duan, "A QRS Complex Detection Algorithm Based on Mathematical Morphology and Envelope," in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, Shanghai, China, Jan. 2005, pp. 4654–4657.

[26] F. Zhang and Y. Lian, "QRS Detection Based on Multiscale Mathematical Morphology for Wearable ECG Devices in Body Area Networks," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 4, pp. 220–228, Aug. 2009.

[27] V. Afonso, W. J. Tompkins, T. Nguyen, and S. Luo, "ECG Beat Detection Using Filter Banks," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 192–202, Feb. 1999.

[28] H. Dinh, D. Kumar, N. Pah, and P. Burton, "Wavelets for QRS Detection," in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, Istanbul, Turkey, Oct. 2001, pp. 1883–1887.

[29] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-Network-Based Adaptive Matched Filtering for QRS Detection," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 4, pp. 317–329, Apr. 1992.

[30] L.-Y. Shyu, Y.-H. Wu, and W. Hu, "Using Wavelet Transform and Fuzzy Neural Network for VPC Detection from the Holter ECG," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1269–1273, Jul. 2004.

[31] D. T. Kaplan, "Simultaneous QRS Detection and Feature Extraction using Simple Matched Filter Basis Functions," in *Proceedings of IEEE Computers in Cardiology*, Chicago, IL, US, Sep. 1990, pp. 503–506.

[32] J. Cox, H. Fozzard, F. M. Nolle, and G. Oliver, "AZTEC, A Preprocessing System for Real-Time ECG Rhythm Analysis," *IEEE Transactions on Biomedical Engineering*, vol. 15, no. 2, pp. 128–129, Apr. 1968.

[33] J. P. Abenstein and W. J. Tompkins, "A New Data-Reduction Algorithm for Real-Time ECG Analysis," *IEEE Transactions on Biomedical Engineering*, vol. 29, no. 1, pp. 43–48, Jan. 1982.

[34] F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig, "Principal Component Analysis in ECG Signal Processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 1–21, Feb. 2007.

[35] H. Lee and K. M. Buckley, "ECG Data Compression Using Cut and Align Beats Approach and 2-D Transforms," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 5, pp. 556–564, May 1999.

[36] D. Zordan, B. Martinez, I. Villajosana, and M. Rossi, "On the Performance of Lossy Compression Schemes for Energy Constrained Sensor Networking," *ACM Transactions on Sensor Networks*, vol. 11, no. 1, pp. 15:1–15:34, Nov. 2014.

[37] N. Maglaveras, T. Stampkopoulos, K. Diamantaras, C. Pappas, and M. Strintzis, "ECG pattern recognition and classification using non-linear transformations and neural networks: A review," *International Journal of Medical Informatics*, vol. 52, no. 1–3, pp. 191–208, Oct. 1998.

[38] C. C. Sun and S. C. Tai, "Beat-based ECG compression using gain-shape vector quantization," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1882–1888, Nov. 2005.

[39] J. Cardenas-Barrera and J. Lorenzo-Ginori, "Mean-Shape Vector Quantizer for ECG Signal Compression," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 1, pp. 62–70, Jan. 1999.

[40] P. Soh, G. Vandenbosch, M. Mercuri, and D.-P. Schreurs, "Wearable Wireless Health Monitoring: Current Developments, Challenges, and Future Trends," *IEEE Microwave Magazine*, vol. 16, no. 4, pp. 55–70, May 2015.

[41] A. Johansson, P. Öberg, and G. Sedin, "Monitoring of Heart and Respiratory Rates in Newborn Infants Using a New Photoplethysmographic Technique," *Journal of Clinical Monitoring and Computing*, vol. 15, no. 7–8, pp. 461–467, Dec. 1999.

[42] K. Chon, S. Dash, and K. Ju, "Estimation of Respiratory Rate From Photoplethysmogram Data Using Time–Frequency Spectral Estimation," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 8, pp. 2054–2063, Aug. 2009.

[43] M. Danieletto, N. Bui, and M. Zorzi, "RAZOR: A Compression and Classification Solution for the Internet of Things," *Sensors*, vol. 14, no. 1, pp. 68–94, Jan. 2014.

[44] N. M. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of First-Derivative Based QRS Detection Algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 478–484, Feb. 2008.

[45] M. Elgendi, "Fast QRS Detection with an Optimized Knowledge-Based Method: Evaluation on 11 Standard ECG Databases," *PLoS ONE*, vol. 8, no. 9, pp. 1–18, Sep. 2013.

[46] E. Keogh and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, Mar. 2005.

[47] S. Salvador and P. Chan, "Toward Accurate Dynamic Time Warping in Linear Time and Space," *Journal of Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

[48] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, US: Kluwer Academic Publishers, 1991.

[49] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[50] S. C. Tai, "Adaptive Sampling of One Dimensional Digital Signal," R.O.C. Patent 083 501, Jan., 1997.

[51] J. Karhunen and J. Joutsensalo, "Generalizations of Principal Component Analysis, Optimization Problems, and Neural Networks," *Neural Networks*, vol. 8, no. 4, pp. 549–562, Jan. 1995.

[52] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[53] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, Dec. 2010.

[54] E. Candès and J. Romberg, "Sparsity and Incoherence in Compressive Sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Nov. 2007.

[55] R. Baraniuk, "Compressive Sensing [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[56] E. Candès and J. Romberg, "ℓ_1-MAGIC," Matlab routines for solving CS problems, 2005. [Online]. Available: http://statweb.stanford.edu/~candes/l1magic/

[57] W. Dai and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[58] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A Fast and Accurate First-Order Method for Sparse Recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, Jan. 2011.

[59] 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, Feb. 2013.

[60] Z. Zhang and B. D. Rao, "Extension of SBL Algorithms for the Recovery of Block Sparse Signals with Intra-Bock Correlation," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009–2015, Apr. 2013.

[61] A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics: the Discrete Wavelet Transform*. Springer, 2001.

[62] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous Sparse Approximation via Greedy Pursuit," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, US, Mar. 2005.

[63] S. Lee, J. Kim, and J.-H. Lee, "A Real-Time ECG Data Compression and Transmission Algorithm for an e-Health Device," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2448–2455, Sep. 2011.

[64] H. Lee and K. M. Buckley, "Heart Beat Data Compression Using Temporal Beats Alignment and 2-D Transforms," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Pacific Grove, CA, US, Nov. 1996, pp. 1224–1228.

[65] V. Lukin, M. Zriakhov, A. Zelensky, K. Egiazarian, and A. Varri, "Lossy Compression of Multichannel ECG Based on 2-D DCT and Pre-processing," in *Proceedings of International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, Lviv-Slavske, Ukraine, Feb. 2008, pp. 159–162.

[66] V. Allen and J. Belina, "ECG Data Compression Using the Discrete Cosine Transform (DCT)," in *Proceedings of Computers in Cardiology*, Durham, NC, US, Oct. 1992, pp. 687–690.

[67] M. Alam and N. Rahim, "Compression of ECG Signal Based on its Deviation from a Reference Signal Using Discrete Cosine Transform," in *Proceedings of the International Conference on Electrical and Computer Engineering*, Dhaka, Bangladesh, Dec. 2008, pp. 53–58.

[68] E. Anas, M. Hossain, M. Afran, and S. Sayed, "Compression of ECG Signals Exploiting Correlation between ECG Cycles," in *Proceedings of*

the International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, Dec. 2010, pp. 622–625.

[69] R. Benzid, F. Marir, A. Boussaad, M. Benyoucef, and D. Arar, "Fixed Percentage of Wavelet Coefficients to Be Zeroed for ECG Compression," *Electronics Letters*, vol. 39, no. 11, pp. 830–831, May 2003.

[70] ARM The Architecture for the Digital World, "ARM Cortex-M4 Processor," 2015. [Online]. Available: http://www.arm.com/products/processors/cortex-m/

[71] 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, May 2013.

[72] Texas Instruments, "CC2451: 2.4 GHz Low Energy and Proprietary System-on-Chip," 2015. [Online]. Available: http://www.ti.com/product/cc2541

[73] M. Blanco-Velascoa, F. Cruz-Roldána, J. I. Godino-Llorenteb, J. Blanco-Velascoa, C. Armiens-Aparicioa, and F. López-Ferreras, "On the Use of PRD and CR Parameters for ECG Compression," *Elsevier Medical Engineering & Physics*, vol. 27, no. 9, pp. 798–802, Nov. 2005.

[74] M. Saeed and M. Villarroel and A.T. Reisner and G. Clifford and L. Lehman and G.B. Moody and T. Heldt and T.H. Kyaw and B.E. Moody and R.G. Mark, "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database," *Critical Care Medicine*, vol. 39, no. 5, pp. 952–960, May 2011.

[75] Y. Zigel, A. Cohen, and A. Katz, "ECG Signal Compression Using Analysis by Synthesis Coding," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, pp. 1308–1316, Oct. 2000.