A noble double dictionary based ECG Compression Technique for IoTH

Jia Qian *Member, IEEE,* Prayag Tiwari *Member, IEEE,* Sarada Prasad Gochhayat *Member, IEEE,* Hari Mohan Pandey *Senior Member, IEEE,*

***Abstract*—Internet-of-things (IoT) health-care system monitors a patients' condition and takes preventive measures in case of an emergency. Electrocardiogram (ECG) that measures the electrical activity of the heart is one of the important health indicators. Thanks to the wearable technology, nowadays, we can even measure the ECG using smart portable devices and send via a wireless channel. However, this wireless transmission has to minimize both energy and memory consumption. In this paper, we propose CULT - an ECG compression technique using unsupervised dictionary learning. Our method achieves a high compression rate due to the essence of dictionary learning and is immune to the noise by integrating Discrete Cosine Transformation. Moreover, it continuously expands the dictionary when the unseen pattern occurs and refines the dictionary when new input arrives, by imposing the double dictionary scheme. We show that our method has a better performance by comparing it with the other existing approaches.**

***Index Terms*—ECG, compression, Vector Quantization, Dictionary Learning, IoT healthcare.**

I. Introduction

INTERNET of things (IoT) health-care systems promise to provide effective health services by continuously monitoring patients' condition and taking preventive measures in case of emergencies [1], [2], [3]. One of the important parameters which need continuous monitoring is electrocardiogram (ECG). ECG measures the electrical activity of the heart over a period of time by placing electrodes on the skin. These electrodes detect the tiny electrical changes on the skin that originate from the heart muscle's electro-physiologic pattern of depolarization and re-polarization during each heartbeat. Thanks to wearable technology, nowadays, we can even measure the ECG using smart portable devices. When wearable is used, sending the ECG records remotely via the wireless channel is very useful. However, this wireless transmission is also challenging and leads to some issues in terms of **energy**, **noise interference** especially when the wearable devices are battery operated. As a solution to this, we advocate compressing the ECG records directly on the wearable device, through some lossy compression technique, and transmitting this compressed representation. At the receiving end, the compressed signal is translated back into the original one using a decompression technique. To achieve high compression rates, we may trade some representation accuracy for high energy efficiency.

In general, the compression techniques can be classified into two categories, *lossy* and *lossless*. While lossless compression techniques guarantee the signal at the decompression side is reconstructed without error but with low compression ratios, lossy compression techniques may reach much higher compression ratios but with the cost of reconstruction accuracy. In this work, we focus on lossy compression techniques. One of the lossy compression classes is a transformation-based lossy compression, where an encoder processes the input signal via an invertible transformation, i.e., transforming the input signal into a new domain (whose dimensionality smaller than the original space); and the decoder reconstructs it by applying the inverse transform. This class includes the Karhunen-Loeve transform (KLT) [4], [5], the Fourier transform (FT) [6], [7], the cosine transform (CT) [8], and the wavelet transform (WT) [9], [10], [11]. The algorithms in this category may achieve relatively high compression ratios but their downside is the high computational complexity, which often implies high energy consumption at both compressor and transmitter.

The other class of lossy compression is known as parameter extraction-based lossy compression, which generated great interest from the research community in the past decade due to new Artificial Neural Network (ANNs) designs [12], [13], [14] (e.g., autoencoder[15]). Suitable ANNs can be used to solve the VQ problem at the compressor. Vector quantization (VQ) [16], [17], [18], for example, is used to construct suitable dictionaries that are then exploited for compression. Unlike transform methods, VQ consists of processing the input time series to obtain some kind of knowledge, like the distribution, the recurrent pattern (motifs) or the structural property of the signal, and utilizes these to get compact and accurate signal representations. Some representative algorithms are Gain-Shape Vector Quantization (GSVQ) [19], [20], Lightweight Lossy Compression (LTC) [21], [22] etc. This is still a field with limited investigation with respect to others. The proposed ECG compressor that we design in this work falls within this

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 764785, FORA-Fog Computing for Robotics and Industrial Automation.

J. Qian is with Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Lyngby, Denmark (e-mail: jiaq@dtu.dk). This work is based on my Master thesis in University Of Padova.

P. Tiwari is with the Department of Information Engineering, University of Padova, Padova, 35131, Italy. (e-mail: prayag.tiwari@dei.unipd.it)

S. P. Gochhayat is with the Virginia Modeling Analysis and Simulation Center, Old Dominion University, Norfolk, VA, 23435, USA. He is also with the Department of IT, Manipal University Jaipur, India. (e-mail: sarada1987@gmail.com)

H. M. Pandey is with the Department of Computer Science, Edge Hill University, Ormskirk, L39 4QP, United Kingdom. (e-mail: pandeyh@edgehill.ac.uk)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974678, IEEE Internet of Things Journal

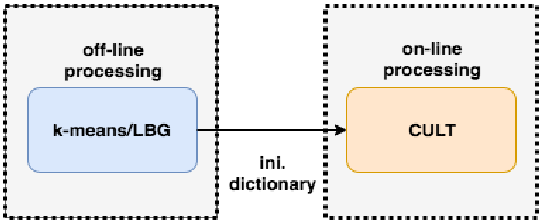


Fig. 1. high-level diagram.

class.

The vast majority of the above mentioned methods do not guarantee the perfect reconstruction of the signals. Although it is necessary the reconstruction error between the original and the reconstructed signals should be very small, there is no automatic way to assure that the distortion in the reconstructed signal will not affect the morphological features of ECG. The closest work compared to our work is based on time adaptive self-organizing map (TASOM) model [23]. The TASOM-based scheme uses a single dictionary, and whenever a new pattern is measured, it is used to train the *only* dictionary it has, which is concurrently used to compress. So the problem in TASOM arises, as it distorts a trained dictionary in order to represent a new portion (unseen before) of the state space, and incurs computational burden when the pattern is not recurrent. At the same time, in order not to delay the reconstruction at the receiver, this pattern is immediately needed to be sent.

Based on the above premise, this paper proposes a lightweight compression scheme for ECG signals that will lead to small and acceptable reconstruction errors at the receiver. Basically, the new scheme requires two main phases. The first phase is performed *off-line* and corresponds to building a preliminary dictionary (shown in Figure 1). The k-means [24], [25] and LBG [26] algorithms are the two candidates to this end. After running the first phase, the so-called **initial dictionary** is available and the problem left is to update it at run-time. Hence, during the second phase (also referred as *online* phase), when the ECG signal is detected, according to the matching level between codeword and signal, we compress it in different ways, more details will be introduced later. This is the main difference with respect to the TASOM algorithm, where each ECG pattern is used to adapt the dictionary, no matter whether the pattern is a proper ECG segment or not. he signal stays in time-series domain unless the segment is transmitted by DCT (or WT), which is different from [27] that the feature vector is composed in frequency domain. Our main contribution can be sketched as follows:

- We propose a double-dictionary scheme that is robust to noise and the destroy of input outliers.
- We aim an overall good method by considering the compression rate, reconstruction accuracy, and energy consumption at the same time.
- Our method is aware of the statistics change, namely, it has the capability of learning new distribution.
- Our method is robust to the noise and the compression is real-time, no delay.
- It is tunable in terms of accuracy requirements; when a high precision is required, it may achieve it at the price of a smaller compression ratio and vice-verse.

The remainder of this paper is organized as follows. In Section II, we introduce the different mechanisms used to design our proposed solution for ECG compression. And section III is covered the introduction of double-dictionary scheme. Section IV provides detailed experimental results of CULT. Section V concludes the paper.

II. Proposed Dictionary based ECG Compression Scheme

The general idea behind the proposed ECG compression algorithm is to utilize a dictionary to represent recurring patterns (also referred to as motifs). During the initial setup, the sender generates the dictionary and sends it to the receiver. When the sender receives a new ECG segment, it checks the closest codeword in the dictionary and selects it as the best-match. The rationale is then to transmit the index associated with the best matching codeword in place of the entire ECG segment, thus achieving a high compression rate (as an index takes much fewer bits than the signal). We remark that this entails several challenges: (i) the dictionaries at the sender and at the receiver must be synchronized at all times, (ii) the dictionary should be continuously adapted in order to be well representative of the input space at all times, (iii) the approach should be insensible to artifacts, which should be detected and treated without affecting the learning phase. Here, the problem is that the dictionary accuracy may be compromised. Next, the dictionary is subject-specific. As different subjects have different cardiac cycle periods, the amplitude and duration of the morphological features (P-wave, QRS complex, ST-segment, and T-wave) of the ECG signal of different subjects differ. Hence, a dictionary obtained for a certain subject usually does not represent well for the ECG segments of another person. Hence, the basic idea is to leverage the past temporal patterns from the same subject to build and refine a dictionary that is representative of the ECG produced by the subject, obtaining a subject-specific-based algorithm.

The proposed algorithm is a VQ-based, online, dictionary-based algorithm. As a new ECG input segment is measured, two tasks are carried out: (i) to figure out how the current dictionary can be exploited to compress the segment and (ii) updating and reshaping the dictionary so that it will learn to represent the new input. Figure 2 shows the overall flow our proposed CULT model. It consists of 3 phases: preprocessing, off-line preliminary dictionary building, and online update of the initial dictionary at runtime. Each step is described in detail in the following subsections.

A. Preprocessing

In the preprocessing step, baseline wanders and high-frequency noise were removed from ECG signals. These were removed by passing each signal through a digital third-order Butterworth high-pass filter with a cutoff frequency of 0.5 Hz (for baseline wandering) and a third-order Butterworth low-pass filter with a cutoff frequency of 70 Hz (for high-frequency noise). The filters were applied both in forward and backward

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974678, IEEE Internet of Things Journal

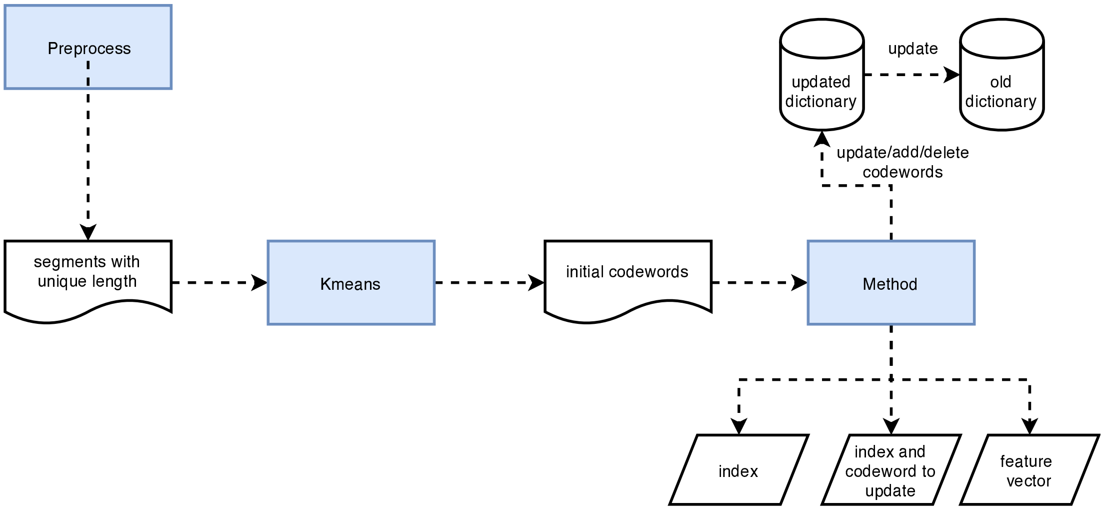


Fig. 2. Overall flow of the proposed dictionary learning method for ECG compression. The raw signal is primarily applied, after that the normalized segments of signal is feed into kmeans. After that, it outputs the initial dictionary with k codewords. Finally, we implement double-dictionary method.

direction to get zero-phase distortion. The next step involves the detection of QRS complex, for which we adopted the optimized knowledge-based method proposed in [28]. Then, the signal was normalized by calculating the z-score in order to remove the amplitude differences between different segments.

B. Offline Training Phase

In this phase, we use *k*-mean algorithm [24] to create the initial dictionary by clustering the signal segments. Despite suboptimal performance of *k*-mean, we used it mainly because of its modest complexity. Setting *k*, i.e., the number of centroids (the initial size of dictionary) and providing the algorithm with a sufficient number of training examples, that in our case are ECG segments, we exploit *k*-means to obtain the initial *k* centroids. These correspond to the initial *k* codewords of the initial dictionary. Both the gradient descent algorithm (GDA) and the stochastic gradient algorithms (SGD) are possible for the construction of the preliminary dictionary. GDA updates the model based on the whole training dataset, which leads to a *deterministic* output. Namely, for a convex problem, every time we run GDA for a given training set, it converges to the same result. While SGD is stochastic, we are no longer using the entire training set at once, instead picking one or more examples (mini-batch algorithm [29]) at a time, each time it returns a different optimum due to the random order of training data. However, when the number of data set is huge, SGD would be a good choice, since it runs much faster. In our case, we decided to build it by GDA since the size of our data set is modest. For the more complicated problem, we can use Gaussian Mixture Model, known as the soft k-means, normally it outperforms k-means, but with more computation demanding.

C. Online Training Phase

The online training phase is critical in our approach, making the method aware of the new distribution if input signal and robust to the outliers. In online phase, the approach should meet the following two requirements:

- it should adapt to differing signals, to effectively deal with noisy patterns (artifacts) and achieve all of this without knowing the signal statistics; and
- it should be tunable in terms of accuracy requirements; when high precision is required, it may achieve it at the price of a smaller compression ratio and vice-verse, and high compression will be achieved at the price of larger representation errors.

To accomplish these requirements, we identified [30] as a good option to adapt the centroids, i.e., update the dictionary codewords at runtime. It can be considered as a dynamic neural network, initially with N_i neurons, each of which has an entangled vector of weights C_{N_i}, corresponding to synaptic weights. During the procedure, the number of neurons varies by adding and removing operations and the weights vector keeps updating. Such that, we can prune the nodes and connections that are not commonly used as the representatives and expand the network by adding the new nodes when new statistics occur.

D. Dynamic Neural Network

1) Input the preprocessed ECG segment (See Section II-A),
2) Find the nearest and the second-nearest nodes, referred to here as N_1 and N_2, respectively,
3) Increase the age of all the edges emanating from N_1,
4) Every codeword has a counter, which records the distance accumulated between itself and the inputs. Update the counter of N_i

$$counter(N_1) = \|C_{N_1} - X\|$$

where X is the current ECG input, C_{N_1} is the codeword corresponding to node N_1,

5) Update C_{N_1} using the learning rate θ_b and N_i's neighbors using learning rate θ_n

$$C_{N_1} = C_{N_1} + \theta_b(X - C_{N_1})$$

$$C_{N_i} = C_{N_i} + \theta_n(X - C_{N_i})$$

where i represents all the neighbours of N_1.

6) If N_1 and N_i are connected, set the age of this edge equal to zero, if not, connect them, and set the age of the edge to zero.

7) After every λ inputs, remove edges with an age greater than A_{max}, if after removing some node (codeword) some nodes become isolated, remove them as well.

8) After every λ inputs, we split the edge between the node (codeword) with the highest counter and its neighbor with the highest counter among all its neighbors:

- Let us denote the node (codeword) with the highest counter as h, and its neighbor with the highest counter as q
- Insert a new codeword n such that its synaptic weight vector is:

$$C_n = 0.5(C_h + C_q)$$

- Remove the original edge between h and q, connect q and n, connect h and n, set the ages of these links to zero,
- Reduce the counters of h and q multiplying them by a constant $0 < \alpha < 1$. Initialize, the counter of n as the new counter of h.

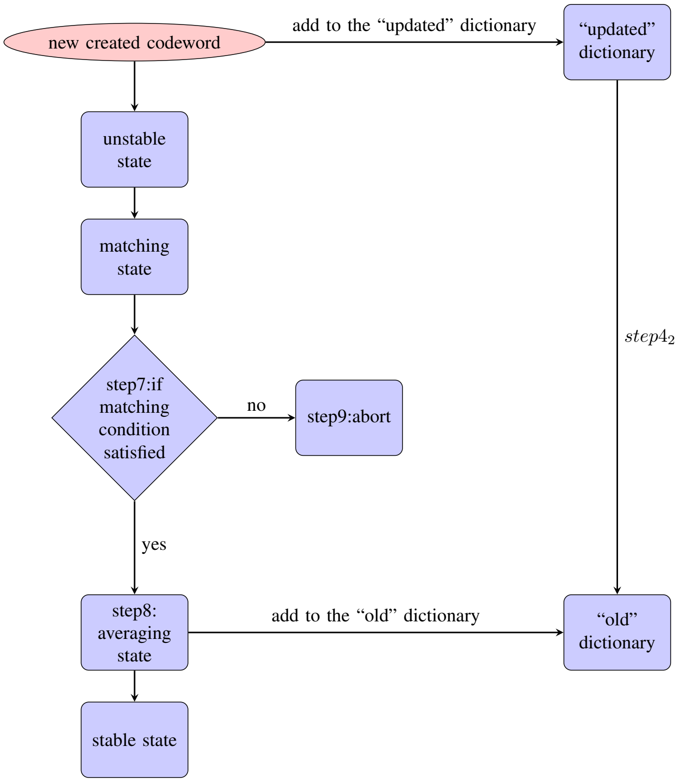


Fig. 3. Schematic illustration of double-dictionaries method. The recent created codeword in "updated" dictionary will be labeled as "unstable state" and pass it to matching state. If the criterion is not met, we abort it, otherwise, it moves to averaging state and finally stable state, and add to "old" dictionary.

III. A double-dictionary Scheme

In this section, we present CULT a new ECG signal compression approach, it is an integrated scheme that seeks robustness and adaptiveness at the same time. CULT is empirically designed, as the noisy signal is unavoidable in the wireless environment. In the new approach, based on the dynamic neural network, we added a double-dictionary mechanism, which is key to come up with a flexible solution. Basically, we have two dictionaries, the first one called "old" dictionary that is stored at both encoder and decoder sides (a copy of initial dictionary), whereas the second one called "updated" dictionary is only stored at the encoder side. Initially, both are the same, but as new input segments are measured the "updated" dictionary keeps updating.

The most important reason here is that once we approximate well a portion of the data space, we consolidate that knowledge into the so-called "old" dictionary. When a previously unseen pattern is measured, this pattern can hardly be represented by the "old" dictionary, with sufficient accuracy. We label this new pattern as an observation state where we check whether this new behavior will become recurrent. We do this through a second dictionary where this new pattern goes through a matching and a subsequent training phase. If the pattern passes the matching phase, which implies that it has become a recurrent pattern and we add it (upon it has gone through some sufficient training) to the consolidated ("old") dictionary. Place it separately, the "old" dictionary encodes the valid distribution according to the previous inputs, it may represent the early inputs data with some certain accuracy. Rather, the "updated" dictionary tries to capture the new portion of the data space/distribution and exclude the outliers generated from the wireless transmission.

In the "updated" dictionary the codewords have two states: stable and unstable. The stable state means that the corresponding codeword has been observed several times and has been used to successfully represent other ECG segments in the past. A codeword in the stable state is represented in both "updated" and "old" dictionaries. And the codeword in the "old" dictionary is used to perform compression, whereas that in the "updated" dictionary is continuously adjusted as new input patterns are arriving. When we refined the codeword in "updated" dictionary, we do not update the corresponding one in "old" dictionary unless the difference between them is considerable. By doing so, we reduce the synchronization cost and make the precision tunable by defining "considerable". In other words, if the codeword in the "updated" dictionary satisfies the error condition (below the error threshold), whereas the corresponding one in the "old" dictionary does not, in this situation, it is mandatory to update the "old" dictionary.

The unstable state indicates that the codeword has not yet been added to the "old" dictionary, which means that it represents an unusual input pattern (not sufficiently representative). When a new input ECG pattern is detected, a new codeword is added to the "updated" dictionary, and label its state as "unstable". This codeword will become stable only if we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2974678, IEEE Internet of Things Journal

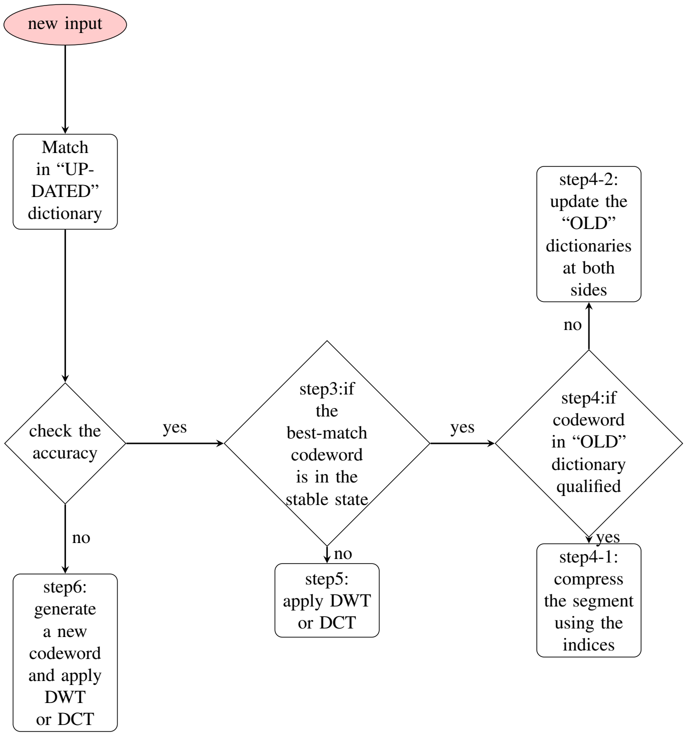

new input
Match in "UP-DATED" dictionary
check the accuracy
yes
no
step6: generate a new codeword and apply DWT or DCT
step3:if the best-match codeword is in the stable state
yes
no
step5: apply DWT or DCT
step4:if codeword in "OLD" dictionary qualified
no
step4-2: update the "OLD" dictionaries at both sides
yes
step4-1: compress the segment using the indices

Fig. 4. Overall approach. When a new input arrives, we firstly match it with ”updated” dictionary, if the best-match codeword doesn’t offers sufficiently high accuracy, we add new codeword to the ”updated dictionary” and compress the input by DWT or DCT. Otherwise, we check the state of the best-match one, if it’s stable and existed in ”old” dictionary, we deliver its corresponding index; if if it’s stable but not existed in ”old”dictionary, we update ”old dictionary” and deliver its corresponding index. While, if it’s not stable yet, we again compress it by DWT or DCT.

observe that several ECG segments are matched into it within a certain time period. If this happens the codeword becomes ”stable” and is copied to the ”old” dictionary. The reason here is some outliers are unavoidable, especially with wearable devices artifact may frequently occur and these do not match any valid ECG segment. In the matching phase, we check if the newly added codeword is ”hit” by the inputs during some specific time interval, if not, with high probability it is an outlier or at least a less-representative codeword with respect to others. How likely the codeword will pass the matching test also depends on the threshold, for example, if we set a small error threshold (i.e., we require high accuracy), even the new generated codeword is not an outlier, it may not pass the matching phase, the majority of the inputs will then exit from step 6 in Figure 4. If the codeword does not pass the matching phase, it will be simply removed.

DCT and DWT are the candidate algorithms for step 2 and step 3 (see figure 4). Numerical results reveal that DCT provides better reconstruction performance and, for this reason, this transform has been selected as a part of the new approach. To choose the DCT coefficients, we kept adding coefficients in each subsequent iteration, and transform from frequency domain to input domain, and compare the RMSE between the original segment and the one reconstructed from the selected coefficients. The number of coefficients to be added is adaptive and depends on energy consumption. If this number is high, fewer iterations will take place, implying a smaller computational cost, and vice-versa.

Presumably, we have enough training data, the dictionary has been well constructed, and we set a proper error threshold Err_{th} for the later inputs. Upon these conditions, if the RMSE of a new input stays beyond Err_{th}, most likely this input is an outlier or some certain pathological signal. For the encoder, it can not precisely distinguish them immediately. If we do not want any delayed reconstruction (i.e. the delay is caused by the examination of this input), encoding it through DCT or DWT is a good choice. By changing the number of features, we may decide the accuracy of that piece of signal. After that, we add the new input as a new codeword in the ”update” dictionary and go through the checking process. Also, in the future we may explore a further avenue: instead of directly labeling the outlier as a new codeword, maybe, we can have another dictionary, where we have the codewords representing all the frequent cardiovascular diseases. If unfortunately one of them hit, we are aware that it is a pathological signal immediately. The implementation of the double-dictionary scheme is shown in Figure 3.

After presenting the Dynamic Neural Network and the double-dictionary scheme, next we introduce how to combine them together. The framework is described below, and the overall flow diagram is shown in Figure 4.

1) Upon collecting a new ECG segment X, firstly find the best match in the “updated” dictionary according to the shortest Euclidean distance:

$$index^{*} = argmin_{i=1,2,\dots,k}\|X - C_i\|^2$$

X: input
C_i: the i-th codeword

2) Check if the distance is smaller than the preset threshold Err_{th}.
   a) if yes, go to step 3
   b) if no, go to step 6
3) Check the state of the best-matching codeword
   a) if stay stable, go to step 4
   b) if stay unstable, go to step 5
4) According to the $index^{*}$, check whether the Euclidean distance between the codeword in the ”old” dictionary and the current input X is also smaller than threshold Err_{th}
   a) if it is, encoder side transform only the index, offset, gain, the decoder side reconstructs the signal using them,
   b) if it is not, the encoder transforms the codeword in the ”updated” dictionary to the decoder and updates its ”old” dictionary, the decoder updates the ”old” dictionary as well and reconstructs the signal by the new-arrival codeword. Note the ”old” dictionaries in both sides are kept consistent.
5) Since the best-matching codeword remains in the unstable state, this means that it does not exist in the ”old” dictionary, neither at the encoder nor at the decoder side. In this case, we transform the current segment X by DWT or DCT.

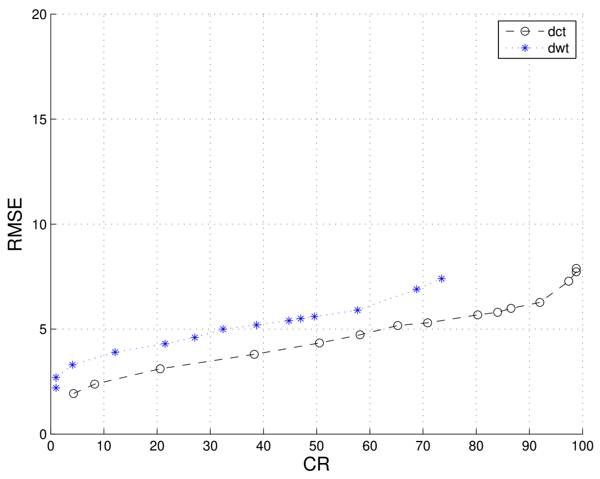


Fig. 5. DCT versus DWT in terms of CR and MRSE.

6) Since no codeword in the "updated" dictionary can represent the current segment X with sufficient accuracy (error with respect to the best matching codeword smaller than Err_{th}), we add a new codeword to the "updated" dictionary, and label the new created codeword as unstable, in the matching phase. The current segment X is compressed using either DCT or DWT.
7) Check if any codeword's training-phase time interval has reached, if it is, go to step 8, otherwise, go to step 10. And check if any codeword's training time has reached, if yes, add it permanently to the "old" dictionary in both sides, otherwise, go to step 10.
8) Check if the codeword has been the best-match for some (predefined) number of input segments, if it is, the codeword passes through the matching phase and moves on to the averaging one. Otherwise, go to step9.
9) Abort it.
10) Continue using Dynamic Neural Network, update its neighbor, check the age scheme, check if some codewords (neurons) have to be added or deleted.

For the new approach, basically every input will "land off" in one of the three categories below:

The first part: the inputs satisfy step2 and step3 in the framework, and locate in the "yes" branch of step3.

The second part: the inputs do not satisfy step2, locate in step6.

The third part: they satisfy step2, but not for step3, finally landing in step5.

Different precision requirements (error thresholds) will cause diverse proportions, it is useful to check the new approach and improve it. If the second part is extremely large, it indicates that the dictionary is not so representative, maybe we need more data to build the initial dictionary or we need to increase the dictionary size.

IV. Data and Experimental Results

In this section, we briefly introduce dataset, implementation and then demonstrate the comprehensive performance of the proposed approach from stressing the different aspects (e.g. energy, error, memory) and try to improve it by the numerical comparison of the tunable parameters (e.g. dictionary size). The proposed approach shows its outstanding property in terms of precision, adaptability.

We use MIT-BIH Arrhythmia Database [31], which is a real dataset measured on different individuals for 48.5 hours. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range.

Implementation: The simulation is carried out on Mac Air laptop with $8G$ ram, under Matlab environment.

A. Performance Matrix

Before we dive into the performance inspection, let's firstly define the performance matrix.

Compression Ratio(CR): it represents the compression performance.

$$CR = \frac{\text{number of bits to represent the original signal}}{\text{number of bits to represent the compressed signal}}$$

Root Mean Square Error (RMSE): it provides an assessment of the reconstruction error, i.e., a measure of the distance between the reconstructed signal and the original one.

$$RMSE(\%) = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \hat{X}_i)^2}{N}} \times 100 \ ,$$

where X_i is the i-th signal sample, $\hat{X}_i$ is the i-th reconstructed sample and N is the total number of samples in the signal.

B. Parameter Selection

Here, we observe the RMSE for different values of CR and Energy to select among transformation (i.e., DCT and DWT) domain needed for the proposed work, and feature selection techniques (i.e., LBG and K-mean); and to understand the impact of dictionary size.

*1) **Comparison between DCT and DWT**:* As mentioned earlier, each ECG segment is transformed using DCT or DWT. Thus, we present here the comparison performance of the proposed new approach combined with either DCT or DWT in terms of CR, RMSE, and energy consumption. We compute RMSE and CR with different tuning parameters for dictionaries of fixed size $k = 30$.

From Figure 5, we see that the DCT transform works better than DWT: for the same CR, DCT obtains a lower RMSE. Also, its compression range is wider than DWT, implying that DCT provides a higher flexibility: we may choose a high CR and a low accuracy (high RMSE) or, on the other hand, a high precision (small RMSE) with a lower CR. Figure 6 shows that energy consumption between DCT and DWT. Here we define the energy as the sum of the mathematical operations (addition, minus, multiplication, division) multiply their corresponding CPU cycle number. DCT ends up with a lower energy consumption performance. In fact, for the same amount of energy used, DCT attains a much higher reconstruction accuracy. When we demand a lower RMSE, the required energy of DWT increases faster than DCT. From both the accuracy and the energy standpoints,

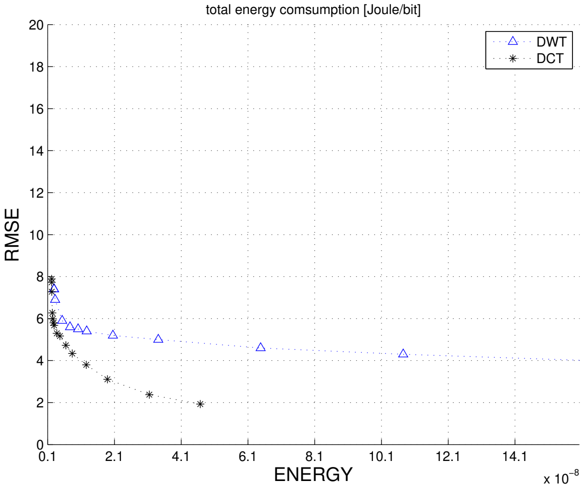


Fig. 6. DCT vs DWT (energy,rmse)

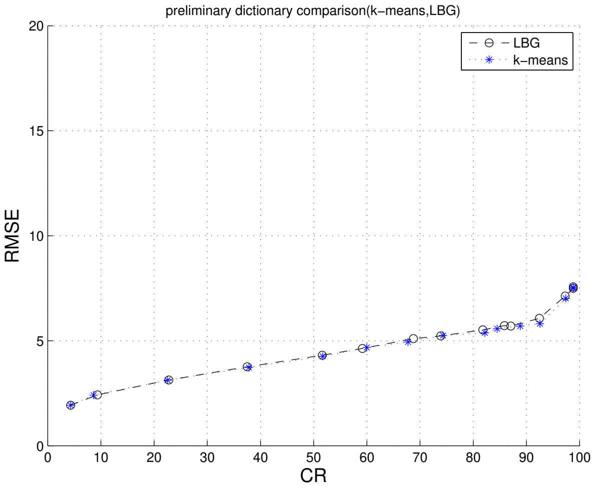


Fig. 7. Effect of two different approaches (LBG or k-means) for preliminary dictionary construction in our method.

We choose DCT as this delivers better performance for both compression and energy usage.

*2) **LBG vs K-means**:* We separately generated the basic (initial) dictionary (codebook) using the LBG and k-means algorithms, and then tested their performance averaging among different subjects. As Figure 7 shows, k-means performs slightly better.

*3) **Impact of Dictionary Size**:* We have mentioned that the variable k (the initial dictionary size) influences the performance of the compression algorithm. Figures 8 and 9 show the change of RMSE energy with respect to variation of CR and consumption performance for k varying from 10 to 100. As shown in Fig. 8 If $k = 10$, the RMSE is bounded between 2.2% and 10%, as k gradually increase, we may achieve a much better result, the smallest RMSE can in fact get even lower than 2%. From Figure 8 we see that when $k = 100$ and CR is very large, the RMSE especially benefits from a large number of codewords in the dictionary. This is actually reasonable, as a larger k basically means that we have more representative codewords learned from previous inputs, and this is expected to lead to higher accuracy. Next, we consider energy consumption and memory problems. From Figure 9, we can see that a larger k implies a smaller energy consumption. We further also see that a larger k also implies a higher accuracy (smaller RMSE). The reason behind is the following: when we have a dictionary with higher precision (more representative codewords), we spend a higher amount of energy in locating the codeword that is closest to the current ECG segment. So the energy increases a little. However, this increment is tiny if compared with the total energy, which is dominated by the energy required to transmit the codeword (DCT) coefficients. This leads to the results of Figure 9, where the energy consumption almost remains the same as we change k while the decrements of RMSE is considerable.

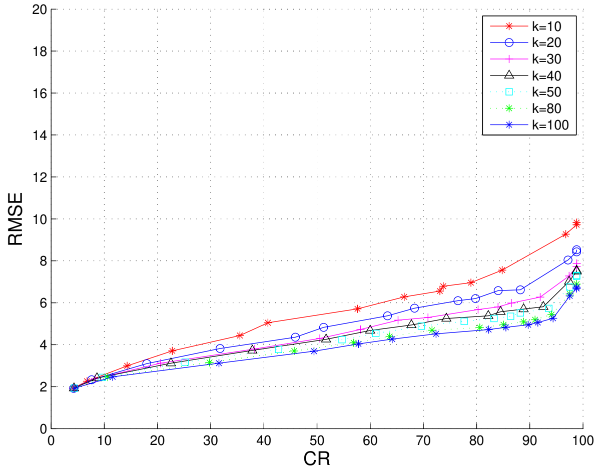


Fig. 8. Compression ratio for different values of k.

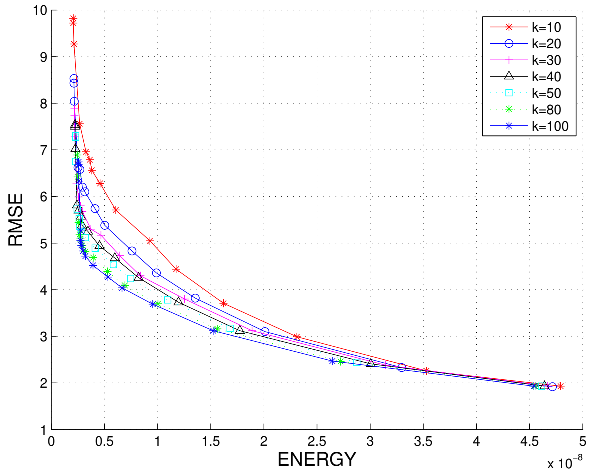


Fig. 9. Energy consumption for different values of k.

Finally, we would like to briefly analyze the memory problem. Assume the length of a codeword is 150 samples (as in our case), every sample is 8bits (one byte), even for case $k = 100$, this means that the total memory is 15kB.

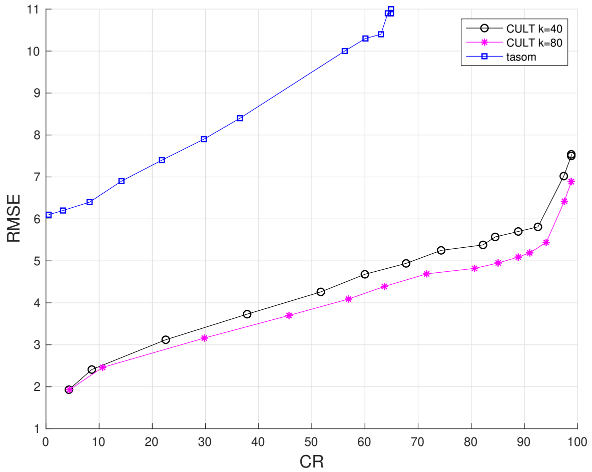


Fig. 10. Compared to TASOM our proposed CULT performs better in terms of RMSE with $k = 80$.

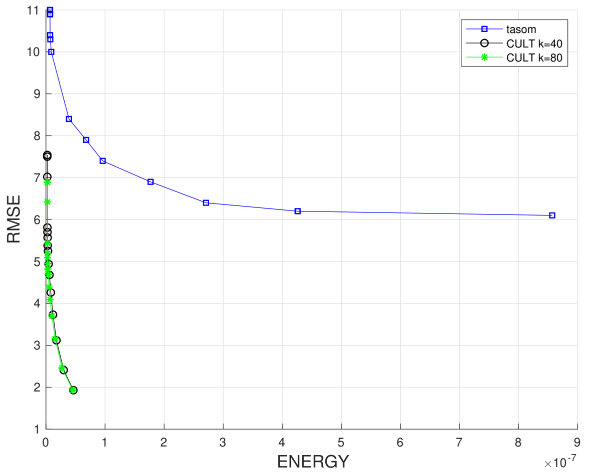


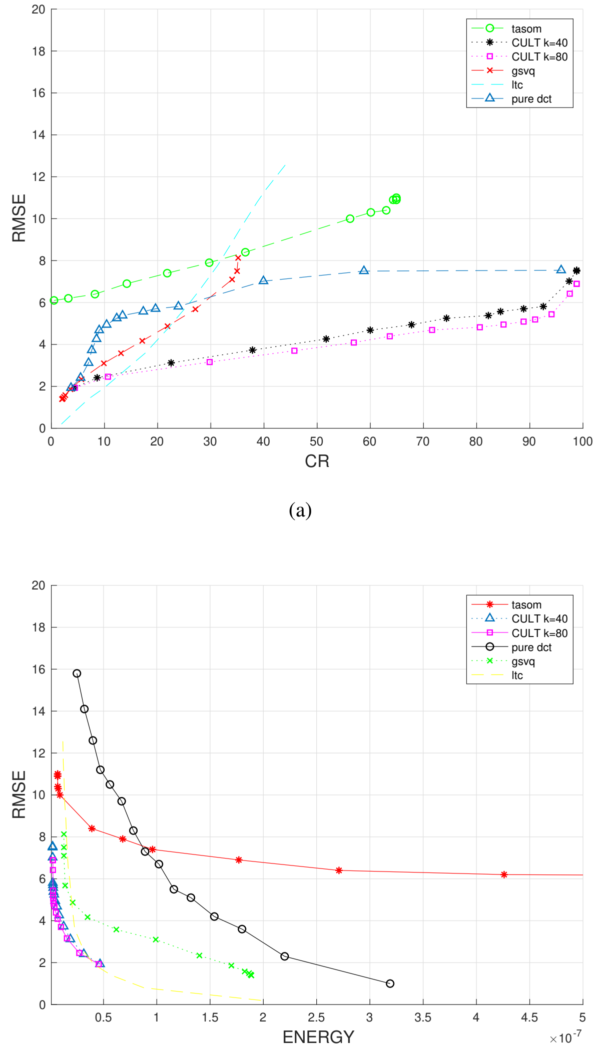
Fig. 11. Comparison of TASOM with our proposed CULT in terms of energy and RMSE.

This figure appears quite reasonable and practical for modern wearable devices.

C. Performance Analysis

In this section, we demonstrate the efficiency of CULT, by comparing it with various compression methods. As TASOM based scheme uses dictionary, we first compare it with TASOM, then we compare with other existing schemes, like GSVQ, ITC, and DCT.

Here, we demonstrate the importance of CULT by comparing CULT with a TASOM-based compressor. The comparison is performed taking account of energy and compression ratio. The TASOM-based scheme uses a single dictionary and, whenever a new pattern is measured, it is used to train the *only* dictionary it has, which is concurrently used to compress. So the problem of TASOM arises, because it has distorted a trained dictionary in order to represent a new portion (unseen before) of the state space. The negative impacts are: (i) retraining takes time, when we measure something new (previously unseen) we need some additional examples to let the dictionary learn and successfully approximate the new patterns, (ii) as the dictionary is reshaped to address the new patterns, it may lose its tracking ability for the state space that was already well covered. Both (i) and (ii) in fact occur when TASOM dictionaries are used and this makes TASOM hardly usable in the presence of sudden changes in the signal statistics or when the signal has artifacts. The new approach adopts the double-dictionary idea and moreover upgrades it by embedding the observation criteria.



(a)

(b)

Fig. 12. Our proposed CULT versus other methods in terms of (a) CR, RMSE and (b) energy.

Figure 10 shows that the new approach leads to a considerable improvement in terms of RMSE. In fact, CULT works well even in the presence of an ECG segment that was never seen before and hasn't been well represented by the old dictionary. The online learning strategy CULT takes charge of retraining the dictionary, while in an attempt to reject outliers and as a consequence that the remained patterns become recurrent. In terms of energy (Figure 11), when RMSE is relatively high (i.e., low compression ratios), CULT and

TASOM have similar performance, whereas TASOM increases its energy consumption more quickly when a low RMSE is required. However, CULT delivers a much higher accuracy (smaller RMSE) when the same amount of energy used. Hence, we realized that the double-dictionary manager plays an important role for CULT.

Figure 12(a) shows the compression performance with respect to other compression algorithms from the literature. The RMSE of CULT is bounded between 1.8% to 7%, The compression ratio locates between 5 and 98. The new compression strategy works much better than other algorithms, LTC has the lowest RMSE, this is a good property, but its RMSE increases rather steeply and its maximum compression ratio is small. By comparing with pure DCT, we can easily know how much CULT improves upon it. Figure 12(b) shows the energy comparison, we see that CULT has a very good performance, its energy expenditure is one magnitude smaller than that of other schemes. Note that the total energy is computed as the sum of processing energy and transmission energy. Here, the energy is expressed in Joule/bit.

D. Morphological comparison

We chose signal 103 from the MIH database as the target ECG signal and compared the reconstructed signal under different methods when RMSE $= 2\%$, and RMSE $= 3\%$. Figures 13-15 show one portion of it when RMSE $= 2\%$, Figures 16-18 when RMSE $= 3\%$. The black line represents the original signal, the red one is the reconstructed one. LTC is pretty coarse, but its linear approximation method disrupts the signal morphology even when the RMSE is rather small, e.g., RMSE $= 2\%$. GSVQ performs much better than LTC, the red line almost follows the trend of the black one, in some regions we may still detect some gap between them when RMSE $= 2\%$. The new approach has the best performance when compared against LTC and GSVQ, achieving the highest CR for any given RMSE. The reconstructed signal by the new approach fits the original signal's shape fairly well.

E. Error comparison

Figures 19 to 21 show the different errors respectively for CULT, LTC, and GSVQ. Among them, LTC shows a pronounced difference between the reconstructed and the original signals. Also, it shows a low-frequency oscillatory behavior, which is superimposed to the reconstructed signal. LTC works by interpolating subsequent points that somewhat follow an almost linear trend by a line segment. If the correlation in the temporal data is high the algorithm is very effective and achieves high CR. LTC is also simple, which means that it has low computational complexity. However, the rigid nature of straight lines leads to a rather great morphological difference between the compressed signal and the original one. This may be a serious problem, as compressing with such an algorithm may lose some of the morphological properties of the signal, which may be important for diagnostic purposes. GSVQ has troubles in accurately matching the QRS complex, as shown in Figure 21. Probably, this can be improved by keeping track of QRS amplitudes. While the difference of the reconstructed

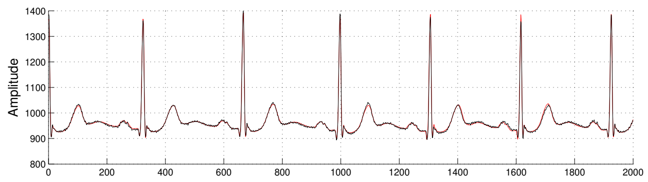


Fig. 13. CULT reconstruction for signal 103, CR= 36, RMSE= 2%

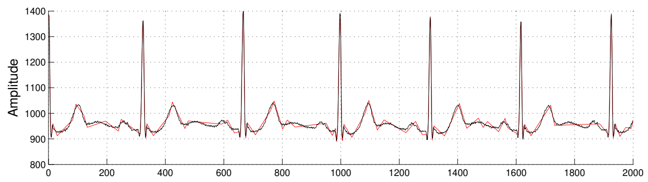


Fig. 14. LTC reconstruction for signal 103,CR=18, RMSE= 2%

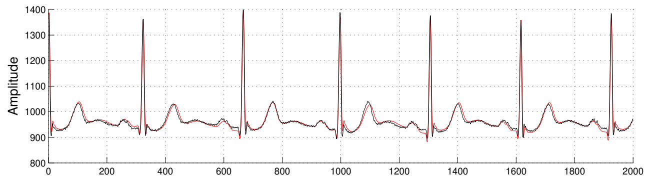


Fig. 15. GSVQ reconstruction for signal 103,CR=29, RMSE= 2%

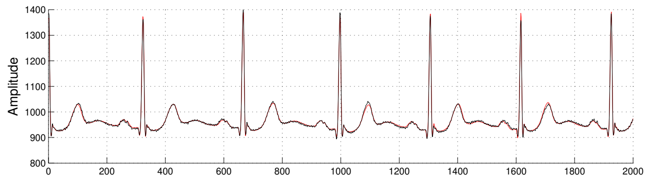


Fig. 16. CULT reconstruction for signal 103,CR=96,rmse=3%

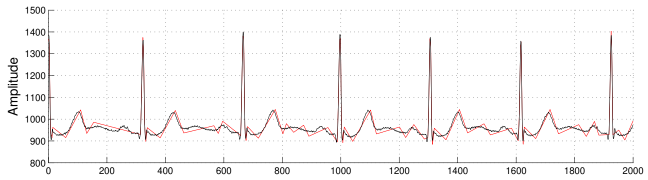


Fig. 17. LTC reconstruction for signal 103, CR=24,rmse=3%

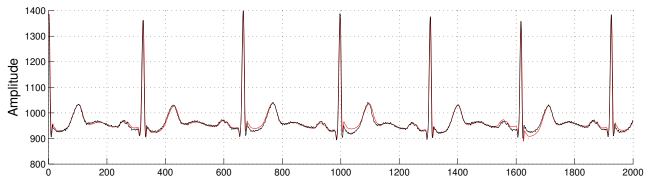


Fig. 18. GSVQ reconstruction for signal 103, CR=35,rmse=3%

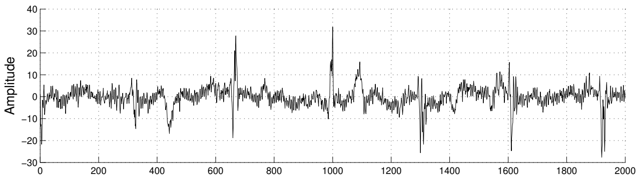


Fig. 19. CULT error for signal 103, CR=96, RMSE=3%

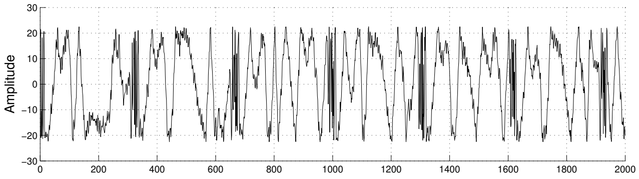


Fig. 20. LTC error for signal 103, CR=24, RMSE=3%

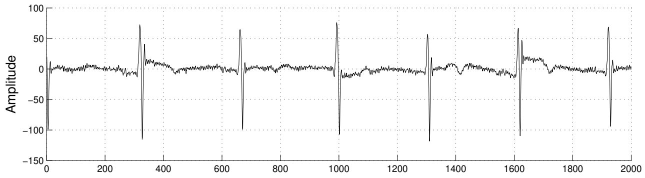


Fig. 21. GSVQ error for signal 103, CR=35, RMSE=3%

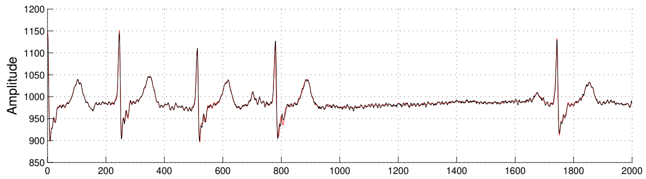


Fig. 22. CULT reconstruction for signal 232, CR= 5.6, RMSE= 3%

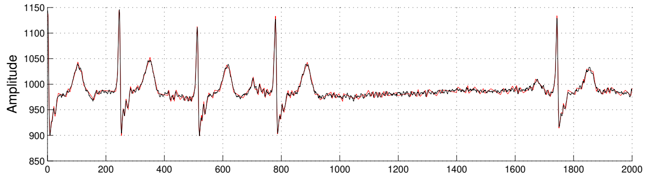


Fig. 23. LTC reconstruction for signal 232, CR= 8.8, RMSE= 3%

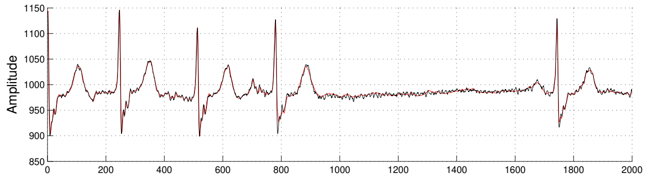


Fig. 24. GSVQ reconstruction for signal 232, CR= 4.6, RMSE= 3%

signal by the new approach is bounded between $+30$ and -30, e.g., for signal 103 with RMSE $= 3\%$, the error mainly concentrates within the range $[-10, +10]$ and the largest error excursion is around the QRS complex. Considering Figure 19, the error resembles the processing of a mixed high-frequency noise signal by a low-pass filter.

In addition to the above results, we show here the adaptiveness of the new algorithm. To this end, signal 232 is an extreme case, as the similarity between subsequent ECG segments is very low, basically they distribute with a great variance within the input space. Figures 22-24 show the reconstructed signal at the decoder side. The new approach provides higher accuracy, especially in the most critical signal regions, the reason for this is attributed to working in the DCT feature space. For such type of signal, the portion handled by DCT is much higher than signals like 103 (which is smoother, more predictable and with a regular behavior). For instance, if we look a the x-axis in the range between samples 900 and 1600, we see that when GSVQ faces the compression of fast-oscillating waves, the reconstructed signal loses the main character of these signal parts. Instead, with the new approach, the information about the upward and downward movements is to a large extent preserved.

V. Conclusion

We proposed a framework for ECG compression in IoT healthcare. As energy and space are scares in IoT, the proposed compression scheme achieves both energy and space improvement by using the dictionary. We compress the signal by detecting recurrent patterns, matching the current(old) dictionary. We observed picking a proper preliminary dictionary size k is the very first important as it affects the algorithm's performance with respect to RMSE, CR, and even energy consumption. We treated a new ECG pattern cautiously when that is unable to match with the existing dictionary as using it to update the current dictionary may actually degrade its representation accuracy. Hence, this pattern is added to a new dictionary where it will be kept under observation, going through an assessment phase. At the same time, in order not to delay the reconstruction at the receiver, this pattern is immediately sent and we achieve this through the transmission of a number of DCT coefficients. This amounts to switching from dictionary-based compression to DCT compression for each ECG segment that is not accurately tracked by the current dictionary. This is the main difference with respect to the TASOM algorithm, where each ECG pattern is used to adapt the dictionary, no matter whether the pattern is a proper ECG segment or not. As we have shown this represents a point of weakness for the TASOM based scheme and a point of strength for our new compression scheme. This amounts to switching from dictionary-based compression to DCT compression for each ECG segment that is not accurately tracked by the current dictionary.

References

[1] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[2] G. Manogaran, R. Varatharajan, D. Lopez, P. M. Kumar, R. Sundarasekar, and C. Thota, "A new architecture of internet of things and big data ecosystem for secured smart healthcare monitoring and alerting system," *Future Generation Computer Systems*, vol. 82, pp. 375–387, 2018.

[3] G. Alfian, M. Syafrudin, M. F. Ijaz, M. A. Syaekhoni, N. L. Fitriyani, and J. Rhee, "A personalized healthcare monitoring system for diabetic patients by utilizing ble-based sensors and real-time data processing," *Sensors*, vol. 18, no. 7, p. 2183, 2018.

[4] A. K. Jain, "A fast Karhunen-Loeve transform for a class of random processes," *IEEE Transactions on Communications*, vol. 24, no. 9, pp. 1023–1029, 1976.

[5] I. F. Jones and S. Levy, "Signal-to-noise ratio enhancement in multichannel seismic data via the karhunen-loeve transform," *Geophysical prospecting*, vol. 35, no. 1, pp. 12–32, 1987.

[6] B. R. S. Reddy and I. S. N. Murthy, "ECG data compression using fourier descriptors," *IEEE Transactions on Biomedical Engineering*, vol. 33, no. 428–433, 1986.

[7] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill, New York, NY, USA, 1986.

[8] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[9] J. Chen, S. Itoh, and T. Hashimoto, "ECG data compression by using wavelet transform," *IEICE Transactions on Information Systems*, no. 1454–1461, 1993.

[10] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

[11] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[12] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, *Distributed representations*. Carnegie-Mellon University, Pittsburgh, PA, USA, 1984.

[13] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[14] A. Němcová, R. Smíšek, L. Maršánová, L. Smital, and M. Vítek, "A comparative analysis of methods for evaluation of ecg signal quality after compression," *BioMed research international*, vol. 2018, 2018.

[15] O. Yildirim, R. San Tan, and U. R. Acharya, "An efficient compression of ecg signals using deep convolutional autoencoders," *Cognitive Systems Research*, vol. 52, pp. 198–211, 2018.

[16] C. P. Mammen and B. Ramamurthi, "Vector quantization for compression of multichannel ECG," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 9, 1990.

[17] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.

[18] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer, 2012, vol. 159.

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

[20] G. R. Canta and G. Poggi, "Kronecker-product gain-shape vector quantization for multispectral and hyperspectral image coding," *IEEE Transactions on Image Processing*, vol. 7, no. 5, pp. 668–678, 1998.

[21] D. D.Testa and M.Rossi, "Lightweight lossy compression of biometric patterns via denoising autoencoders," *IEEE Signal Processing Letters*, 2015.

[22] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," 2004. [Online]. Available: https://escholarship.org/uc/item/6zg2n1rh

[23] H. Shah-Hosseini and R. Safabakhsh, "Tasom: The time adaptive self-organizing map," in *Proceedings International Conference on Information Technology: Coding and Computing*, March 2000, pp. 422–427.

[24] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, no. 281–297, 1967.

[25] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *International Conference on Machine Learning*, vol. 1, 2001, pp. 577–584.

[26] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communication*, vol. 28, no. 1, 1980.

[27] M. Hooshmand, D. Zordan, T. Melodia, and M. Rossi, "Surf: Subject-adaptive unsupervised ecg signal compression for wearable fitness monitors," *IEEE Access*, vol. 5, pp. 19 517–19 535, 2017.

[28] M. Elgendi, "Fast QRS detection with an optimized knowledge-based method: Evaluation on 11 standard ECG database," *PLoS ONE*, no. 9, pp. 1–18, 2013.

[29] J. Konecny, J. Liu, and P. Richtarik, "Mini-batch semi-stochastic gradient descent in the proximal setting," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, 2016.

[30] B. Fritzke, "A growing neural gas network learns topologies," *Ruhr University Bochum, Germany*, 1995.

[31] 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.

Jia Qian (Member, IEEE) received her Master Degree from University of Padova, Italy, Bachelor from Fudan University, China. She is working as a Marie Sklodowska-Curie Researcher and Ph.D. student at Technical University of Denmark, Denmark. She is particularly interested in Distributed Machine/Deep Learning, the applicability of ML on IoT, Edge Computing.

Prayag Tiwari (Member, IEEE) received his Master Degree from National University of Science and Technology "MISiS", Moscow, Russia. He is working as a Marie Sklodowska-Curie Researcher in QUARTZ project (European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 721321) and Ph.D. student at the University of Padova, Italy. He has several publications in top journal and conferences of ACM, IEEE, Elsevier, Springer, etc. His research interests include Machine Learning/Deep Learning, Quantum Theory, Information Retrieval, and IoT.

Sarada Prasad Gochhayat (Member, IEEE) received Ph.D. degree in ubiquitous computing from Indian Institute of Science, Bangalore, India. He worked as a postdoctoral research fellow at the Department of Mathematics, University of Padova, Italy. He was part of the Security and Privacy Through Zeal (SPRITZ) research group, led by Prof. Mauro Conti. His research interests are in the areas of ubiquitous networks, resource management, and computer and network security.

Hari Mohan Pandey (Senior Member, IEEE) received the B.Tech. degree from Uttar Pradesh Technical University, India, the M.Tech. degree from the Narsee Monjee Institute of Management Studies, India, and the Ph.D. degree computer science and engineering from the Amity University, India. He worked as a Postdoctoral Research Fellow with the Middlesex University, London, U.K. He also worked on a European Commission project – Dream4car under H2020. He is a Senior Lecturer with the Department of Computer Science, Edge Hill University, Lancashire, U.K. He is specialized in computer science and engineering and his research area includes, artificial intelligence, soft computing, natural language processing, language acquisition, machine learning, and deep learning.