ECG Monitoring in Wearable Devices by Sparse Models

Diego Carrera[1], Beatrice Rossi[2], Daniele Zambon[2], Pasqualina Fragneto[2], and Giacomo Boracchi[1]

[1] Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, via Ponzio 34/5, 20100 Milano, Italy
diego.carrera@polimi.it, giacomo.boracchi@polimi.it
[2] STMicroelectronics, Via Olivetti 2,
20864 Agrate Brianza, Italy
beatrice.rossi@st.com, daniele.zambon-ext@st.com,
pasqualina.fragneto@st.com

Abstract. Because of user movements and activities, heartbeats recorded from wearable devices typically feature a large degree of variability in their morphology. Learning problems, which in ECG monitoring often involve learning a user-specific model to describe the heartbeat morphology, become more challenging.

Our study, conducted on ECG tracings acquired from the Pulse Sensor – a wearable device from our industrial partner – shows that dictionaries yielding sparse representations can successfully model heartbeats acquired in typical wearable-device settings. In particular, we show that sparse representations allow to effectively detect heartbeats having an anomalous morphology. Remarkably, the whole ECG monitoring can be executed online on the device, and the dictionary can be conveniently reconfigured at each device positioning, possibly relying on an external host.

1 Introduction

In this paper we deal with the problem of monitoring electrocardiogram (ECG) tracings through wearable devices like the Pulse Sensor [1], which is shown in Figure 1 and developed in a joint collaboration between MR&D and STMicroelectronics. Wearable devices have a huge potential in health and fitness scenarios, and in particular in the transitioning from hospital to home/mobile health monitoring. However, to make these devices operational in real-world applications, it is necessary to address relevant machine-learning and data-science challenges. In particular, to provide prompt interaction with the user and prevent massive data-transfer which can spoil their battery life, wearable devices have to autonomously process the sensed data.

In the case of ECG tracings, this processing typically consists in classifying or detecting anomalies in the heartbeats. These tasks are traditionally performed

Fig. 1. Pulse Sensor. The two external electrodes inject the current, while those in the middle read the difference in electric potential. The Pulse Sensor can either analyze onboard, store or transmit the ECG tracings.

by computing expert-based features like those in [6–10], which tend to mimic the criteria clinicians use to interpret ECG tracings. Examples of these features are the ECG values in specific locations of the heartbeat, interval features (e.g., the duration of the QRS, ST-T or QT complex, or the distance between two consecutive peaks, namely the RR distance), and the average ECG energy over these intervals.

Often, expert-based features are combined with data-driven ones, that do not tend to reproduce some clinical evidence but they are directly learned from data [11, 12], possibly by clustering heartbeats [13, 14]. In practice, learning data-driven features boils down to learning a model to represent heartbeats. Since heartbeats of each user are characterized by their own morphology [3] (see the examples of Figure 2), global models are not able to properly describe heartbeats of different users, and lead to poor classification [4] or anomaly detection performance even when trained on large datasets. Therefore, it is convenient to make these models user-specific or at least user-adaptable [5, 6].

Here we focus on data-driven models for learning the morphology that characterize each user heartbeats, and to this purpose, we consider dictionaries yielding sparse representations of heartbeats. Sparse representations are nowadays one of the leading models in image and signal processing [17, 18], and dictionary learning has been successfully used for modeling ECG tracings for anomaly-detection [19, 20] and person-identification [21] purposes. Intuitively, learning a dictionary yielding sparse representations corresponds to learning a union of low-dimensional subspaces where user heartbeats live.

ECG tracings acquired by wearable devices are different from those typically considered in the literature, like the MIT-BIH Arrhythmia Database [15], which contains relatively short segments of good-quality Holter recordings. In the Pulse Sensor, for instance, the electrodes are closer than in an Holter device and these could be mispositioned since they are typically placed by users themselves rather than by clinicians. Moreover, during long-term monitoring, user movements might also cause device displacements. These issues might affect the morphology of heartbeats [16, 3] and have implications on the model used to describe the heartbeats of each user, which are better discussed in Section 2.2.

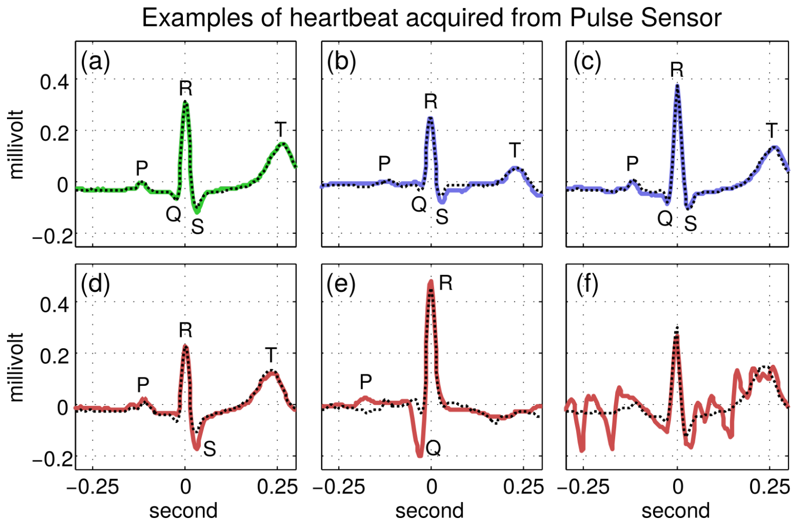

Fig. 2. Examples of heartbeats morphology. The top row (**a**, **b** and **c**) contains heartbeats acquired from user 1 with the Pulse Sensor placed in position 1. In all these heartbeats we depict also the P-waves, the QRS-complexes and the T-waves. The small variations in the morphology of these heartbeats are also due to different heart rates (72 bpm in **a**, 93 bpm in **b** and 77 bpm in **c**). Note also that the morphology remains unaltered over time, since **c** was acquired more than 100 minutes after **a** and **b**. The bottom row (**d**, **e** and **f**) contains heartbeats featuring a different morphology. In particular, **d** reports an heartbeat from the user 1 acquired in position 2 (heart rate of 81 bpm), **e** reports an heartbeat of user 2 (84 bpm) and **c** reports an example of artifact due to movements of user 1. We also report heartbeats reconstructed by the sparse coding with respect to the dictionary learned from user 1 position 1 (dotted lines). Heartbeats in the top row are properly reconstructed (reconstruction errors $r_a = 0.07$, $r_b = 0.15$ and $r_c = 0.10$) since these are from the same user and position. In contrast, the heartbeats in the bottom row show a poor reconstruction quality ($r_d = 0.18$, $r_e = 0.50$ and $r_f = 0.63$). The reconstruction error can be thus used to detect anomalous heartbeats, namely heartbeats that do not feature the morphology characterizing a specific user and electrodes placement.

We here show that dictionaries yielding sparse representations are the right choice for modeling ECG recordings in wearable devices, and that they allow to detect anomalies directly on the device. To this purpose, we consider an anomaly-detection algorithm similar to [22], and study its applicability on the Pulse Sensor. This algorithm is tested over a large dataset of ECG tracings from healthy users, where every heartbeat featuring morphology different from the training ones is considered anomalous. Our experiments show that:

1) Dictionaries yielding sparse representations can successfully describe the *overall variability* in the morphology of heartbeats acquired by wearable devices like the Pulse Sensor. These models do not seem likewise necessary in more controlled situations, as for example in the MIT-BIH Arrhythmia Database, where there

is less variability in the normal heartbeats and anomalies are easier to detect (Section 5).
2) It is possible to detect heartbeats that do not conform the user morphology (i.e., anomalous heartbeats) directly on the device. Indeed, we analyze in detail the computational complexity of a very efficient implementation of the considered anomaly-detection algorithm, and we perform some tests to conclude that this can be reasonably executed in *real-time* on the Pulse Sensor (Section 6).
3) Dictionaries embedded on the Pulse Sensor can provide a *user-adaptable* and *position-adaptable* monitoring solution. In fact, the dictionary learning can be conveniently performed on an external host (e.g., the user's smartphone), requiring only few minutes of ECG tracings as training set. Our experiments also show that this learning phase can tolerate small percentages of heartbeats corrupted by user movements, thus that dictionary learning can be autonomously performed at each device placement (Section 6).

The paper is structured as follows. Section 2 presents the Pulse Sensor and discusses the main challenges of ECG monitoring on wearable devices. The anomaly-detection problem is formulated in Section 3, while we present the considered algorithm in Section 4. Experiments in Section 5, performed on both ECG tracings acquired from the Pulse Sensor and the MIT-BIH Arrhythmia Database, show the that dictionaries yielding sparse representation can effectively model heartbeats and detect those having a different morphology. In Section 6 we study the overall feasibility of this monitoring solution on the Pulse Sensor, while in Section 7 we draw conclusions along with future works.

2 The Pulse Sensor

2.1 Device Description

The Pulse Sensor [1] is a wearable device developed by MR&D in collaboration with STMicroelectronics. It is a battery-powered device, designed for monitoring ECG tracings correlated to other physiological information. In particular, this device continuously acquires, stores and periodically transmits: ECG tracings, measurements of heart rate and breathing rate.

The sensor suite of the Pulse Sensor is made up of one microelectromechanical systems (MEMS) accelerometer, dedicated to estimate both the physical activity and the body position, and four electrodes embedded in a patch (see Figure 1). The outer ones inject AC current with intensity 100 μA and frequency 50 kHz, while the central ones – placed at a distance of 8 cm – read a single-lead ECG (thus a single univariate signal) and a bioimpedance signal.

The main block of electronic components comprises a signal amplifier, three light-emitting diodes (LEDs), a Bluetooth module and a battery. The Bluetooth module connects the Pulse Sensor with a host device (e.g. a smartphone, tablet or a computer) in order to periodically transmit all the acquired signals. The internal battery is a rechargeable Lithium-ion one (3.7 VDC with 350 mAh capacity). The LEDs provide information on the battery charge-status, on the current operational mode of the device (engage, streaming and monitoring) and warnings on the incoming signals. The adopted microcontroller is the STM32F103 which incorporates the ARM® Cortex™-M3 32-bit RISC core operating at a 72 MHz and embeds up to 32 Kbytes of flash memory and up to 10 Kbytes of SRAM.

A peculiarity of this device is modularity, which allows to tailor the sensor suite around specific application requirements. In fact, it is very easy to add new types of electrodes, scaling the software or replacing the microcontroller with a more powerful one as far as this is compatible with the firmware and pinout.

2.2 Issues of ECG monitoring on wearable devices

We here discuss the main issues that makes real-time monitoring of physiological signals particularly challenging in wearable devices like the Pulse Sensor.

Variety of ECG morphology. First of all, during long-time monitoring, the heartbeat morphology might be subject to variations due to changes in the heart rate. This makes ECG tracings acquired by the Pulse Sensor more heterogeneous than ECG tracings acquired in more controlled situations, as for example those of MIT-BIH Arrhythmia Database, which refer to relatively short time intervals. Moreover, the sensing capabilities of the Pulse Sensor are lower than those of devices typically used in clinical trials, since a single ECG tracing is acquired from two electrodes placed at a relatively close distance. The overall variability in the morphology of heartbeats acquired by wearable devices is thus quite large, and difficult to describe.

Computational Constraints. In wearable devices meant for real-time monitoring of physiological signals, sensors continuously acquire data, producing a massive amount of information to be analyzed and possibly stored or transmitted. Needless to say, if the device were periodically transmitting the whole ECG tracings to an host, its battery would be spoiled soon. Data transmission between the wearable and the host can be reduced by enabling the device to autonomously process the sensed data, thus transmitting only the most relevant information, like heartbeats having an anomalous morphology. As such, algorithms used to analyze heartbeats should be compliant with the device computing-capabilities.

Changes in user and device position. ECG tracings do not only depend on the specific user, but also on the specific placement of the ECG electrodes [16, 3]. While this is not an issue when electrodes are placed by clinicians (that at the meantime analyze the ECG tracings) this represents a serious problem in the typical application scenario of wearable devices. In fact, the Pulse Sensor is meant to be positioned by users themselves and, as such, electrodes could be mispositioned, making the model used for automatic analysis unreliable since the heartbeats morphology has changed (see Figure 2.d). The same problem happens during long-term monitoring, when user movements might cause device displacements. Therefore, the model learned on the device has to be easily retrainable every time the device is positioned, without requiring any supervision by an expert clinician. Also, the device configuration should tolerate at least a small fraction of heartbeats affected by user movements.

3 Problem Formulation

We denote by $s\colon \mathbb{N} \to \mathbb{R}$ the ECG tracing which has been uniformly sampled in time, and we assume that the heartbeats have been already segmented e.g., by [29]. We define the i-th heartbeat $\mathbf{s}_i \in \mathbb{R}^p$ as

$$\mathbf{s}_i = \{s(t_i + u) \ : \ u \in \mathcal{U}\}, \tag{1}$$

where $\mathcal{U}$ is a neighborhood of the origin containing p samples, and t_i denotes the sample in the ECG tracing corresponding to the i-th R peak of the ECG tracing. We assume that the normal heartbeats of each wearable-device user are generated by a stochastic process $\mathcal{P}_N$, which characterizes the heartbeats' morphology. Our goal is to learn a model representing the heartbeats morphology; to quantitatively assess the effectiveness of the learned model, we consider the anomaly-detection problem, which is itself of primary concern in ECG monitoring. More precisely, *anomalous* heartbeats are generated by a process $\mathcal{P}_A \neq \mathcal{P}_N$ and exhibits different morphology than heartbeats generated by $\mathcal{P}_N$. Anomalies might be due, for instance, to arrhythmias (as those in the MIT-BIH Arrhythmia Database), movements (as it typically happens in long-term monitoring, e.g. see Figure 2), acquisition errors (which might occur in consumer devices), or simply because these have been acquired from a different user or by changing the electrodes placement (see Figure 2). Anomalies are detected by analyzing each heartbeat $\mathbf{s}_i$ and determining whether it conforms or not the morphology characterizing $\mathcal{P}_N$. When this is not the case, we consider the beat $\mathbf{s}_i$ as anomalous. Since we analyze each beat independently we ignore anomalies that affect, for instance, the heart-rate or that require inspecting multiple heartbeats. We assume only that a training set TR of normal heartbeats is provided, as this allows us to learn a model approximating $\mathcal{P}_N$. We do not require any example of anomalous heartbeats, thus $\mathcal{P}_A$ remains completely unknown. This is a reasonable assumption since normal heartbeats are quite easy to collect and, at least in healthy users, it is enough to record few minutes after having placed the device; in contrast, anomalies are rare and difficult to gather thus the wide range of signals covered by $\mathcal{P}_A$ cannot be properly characterized.

4 The Considered Anomaly-Detection Algorithm

We consider a simple, yet effective, anomaly-detection algorithm that leverages a dictionary yielding sparse representations of the normal heartbeats. In practice, this follows the approach in [22], where a change-detection algorithm was used to monitor rock faces and detect structural changes in fixed-length signals acquired by triaxial MEMS accelerometer. While we use the same model for describing normal data and we analyze the reconstruction error as in [22], we adopt an outlier-detection technique rather than a sequential change-point method for monitoring ECG. This choice better conforms the considered scenario, since the ECG tracings are typically affected by sporadic anomalies rather than permanent changes. In what follows we describe the two main steps of the considered algorithm.

Example of learned dictionary

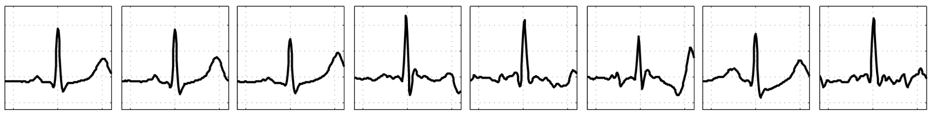

Fig. 3. Atoms of the dictionary learned from the user yielding normal heartbeats in Figure 2. The parameters adopted for the training are $n = 3$, $\kappa = 8$ and $m = 500$.

4.1 Modeling Normal Heartbeats

Our modeling assumption is that the normal heartbeats $\mathbf{s}_i \in \mathbb{R}^p$ of a user are generated from the process $\mathcal{P}_\mathcal{N}$ and can be well approximated by the following linear model

$$\mathbf{s}_i \approx D\mathbf{x}_i \, , \tag{2}$$

where $D \in \mathbb{R}^{p \times n}$ is a matrix called *dictionary* and the coefficient vector $\mathbf{x}_i \in \mathbb{R}^n$ is *sparse* [23]. Sparsity means that $\mathbf{x}_i \in \mathbb{R}^n$ has few of nonzero components, thus in practice that the ℓ^0 "norm" of $\mathbf{x}_i$ is bounded, i.e., $\|\mathbf{x}_i\|_0 \leqslant \kappa$, where $\kappa > 0$ is the maximum number of nonzero coefficients allowed in these representations.

The dictionary D is learned from a training set containing normal heartbeats of a single user. We stack the m heartbeats provided for training in the columns of a matrix $S \in \mathbb{R}^{p \times m}$. *Dictionary learning* consists in solving:

$$[D, X] = \underset{\widetilde{D} \in \mathbb{R}^{p \times n}, \, \widetilde{X} \in \mathbb{R}^{n \times m}}{\arg\min} \|\widetilde{D}\widetilde{X} - S\|_2, \text{ such that } \|\widetilde{\mathbf{x}}_i\|_0 \leqslant \kappa, \;\; i = 1, \ldots, n \tag{3}$$

where the sparsity constraint applies to each column of the matrix $X \in \mathbb{R}^{n \times m}$, which stacks the coefficient vectors of all the heartbeats in S. In practice, (3) can be solved by the KSVD algorithm [24], which alternates the calculation of the dictionary D and the sparse representations of the training heartbeats X.

Thus, the dictionary D is user-specific: its columns, which are referred to as *dictionary atoms*, depict the most relevant morphologies characterizing user heartbeats, as shown in Figure 3. Equation (2) implies that each heartbeat $\mathbf{s}_i$ is approximated by a linear combination of at most κ dictionary atoms.

4.2 Detecting Anomalous Heartbeats

Learning D such that (2) holds for normal heartbeats corresponds to learning a union of low-dimensional subspaces of $\mathbb{R}^p$ where normal heartbeats live. In particular, since the κ atoms can be arbitrarily chosen among the n columns of D, these subspaces can be at most κ-dimensional. The sparse representation $\mathbf{x}_i$ of an heartbeat $\mathbf{s}_i$ can be computed by projecting $\mathbf{s}_i$ on the closest of such subspaces. This problem is referred to as *sparse coding* and it is formulated as

$$\mathbf{x}_i = \underset{\widetilde{\mathbf{x}} \in \mathbb{R}^n}{\arg\min} \|D\widetilde{\mathbf{x}} - \mathbf{s}_i\|_2 \;\text{ such that }\; \|\widetilde{\mathbf{x}}\|_0 \leqslant \kappa. \tag{4}$$

The problem (4) is NP-Hard, and it is typically addressed by greedy algorithms. In particular we here adopt the Orthogonal Matching Pursuit [25], an iterative algorithm which selects the best column of D at each iteration. The OMP can be well implemented in the Pulse Sensor, as discussed in Section 6.

We detect anomalies by assessing whether each heartbeat $\mathbf{s}_i$ to be tested falls in the union of low-dimensional subspaces that characterizes normal heartbeats for a specific user. In particular, we solve (4) and obtain $\mathbf{x}_i$, the coefficients of the closest projection over subspaces of D. Then, we measure the reconstruction error as

$$r_i = \|D\mathbf{x}_i - \mathbf{s}_i\|_2, \tag{5}$$

where $D\mathbf{x}_i$ denotes the linear combination of dictionary atoms that best reconstruct $\mathbf{s}_i$ (the reconstruction of the examples in Figure 2 is reported with dashed lines). The reconstruction error r_i is used to discriminate if s_i is generated by $\mathcal{P}_\mathcal{N}$ or $\mathcal{P}_\mathcal{A}$. In fact, large values of r_i indicate heartbeats that are far from subspaces spanned by columns of D and that as such have a different morphology. Therefore, anomalous heartbeats are detected by determining whether r_i exceeds a suitable threshold $\gamma > 0$, which has to be defined experimentally.

We remark that r_i is a data-driven and user-specific feature, as it is entirely defined from the dictionary D that is learned from the training set without any a-priori information about the heartbeat morphology. Finally, other dictionary-learning and sparse-coding algorithms have been proposed in the literature, and in particular, some of them replace the constrained problems (3) and (4) with their convex relaxation where sparsity is measured by the ℓ^1 norm of the coefficient vectors. These lead to basis pursuit denoising (BPDN) formulation [26]. In the considered settings (see Section 5) these are however more computationally demanding than the OMP, which can be reliably embedded on the Pulse Sensor. It is also worth commenting that, when changing the problems (3) and (4), monitoring the reconstruction error might not be the best option [27].

5 Experiments

In this section we consider two different datasets of ECG tracings: the former was acquired using the Pulse Sensor, the latter is the MIT-BIH Arrhythmia Database [15] that is commonly used in the literature. We consider the algorithm described in Section 4 in a few anomaly-detection scenarios, as a way to quantitatively assess the effectiveness of sparse representations in modeling heartbeats.

5.1 Datasets Description

The *Pulse dataset* contains 20 ECG tracings recorded from 10 healthy users[3] (two tracings per user). The two acquisitions from each user have been performed in different times, repositioning the Pulse Sensor such that the morphology of heartbeats changes. Each ECG tracing lasts from 40 minutes up to 2

[3] The dataset can be made available upon request.

hours and is acquired during normal-life activities, thus the heart rate can significantly vary along the same tracing. Due to motion artifacts or temporary device detachments, these tracings sometimes contain low-quality segments (depicting heartbeats as in Figure 2.e), which have been discarded by an experienced cardiologist with the aid of a commercial software. While these heartbeats are not anomalous from a clinical point-of-view, we exclude them as they do not show the same morphology of others. Possibly, these heartbeats could be removed directly on the Pulse Sensor by monitoring the MEMS recordings. Each ECG tracing is preprocessed as in [7] in order to remove the baseline wander and unwanted power-line and to attenuate high-frequency noise.

The *MIT-BIH Arrhythmia Database* [15] contains 48 ECG tracings lasting around 30 minutes each, that have been extracted from long-term Holter recordings. These segments have been selected by expert cardiologists which discarded the low-quality parts of these traces. Each ECG tracing contains a few arrhythmias, and every heartbeat is provided with annotations by the cardiologists. Both the heart rate and the morphology of normal heartbeats in this dataset are characterized by less variability than in the Pulse dataset.

In all our experiments, we extract heartbeats using a temporal window $\mathcal{U} = [-0.3, 0.3]$ centered in each R-peak, which yield heartbeats having $p = 155$ and $p = 216$ samples in the Pulse Sensor and MIT-BIH dataset, respectively[4].

5.2 Figures of Merit

We consider figures of merit traditionally used to assess the anomaly-detection performance: *i) False Positive Rate* (FPR), namely the percentage of normal heartbeats identified as anomalous and *ii) True Positive Rate* (TPR), namely the percentage of heartbeats correctly identified as anomalous. Since both FPR and TPR depend on the threshold $\gamma > 0$ (see Section 4.2), we consider the Receiving Operating Characteristic (ROC) curve, which are obtained by varying γ and plotting the corresponding TPR against the FPR. An example of ROC curve is provided in Figure 5: the closer the curve to the point (0,1), the better. To get a quantitative assessment of the anomaly-detection performance, we measure the area under the curve (AUC), which for the ideal detector (namely the one having no false positives and no false negatives) is 1.

5.3 Experiments on the Pulse Dataset

Even though ECG recordings from the Pulse dataset were acquired from healthy users and contain no clinical anomalies, we design two anomaly-detection experiments to show that the considered algorithm can effectively detect heartbeats having a different morphology. In particular, we consider as normal (i.e., generated from $\mathcal{P}_N$) heartbeats acquired form a specific user with a specific positioning of the Pulse Sensor. Anomalous heartbeats (i.e., generated from $\mathcal{P}_A$)

[4] Pulse Sensor has a sampling frequency of 256 Hz, while the sampling frequency in the MIT-BIH Arrhythmia Database is 360 Hz.

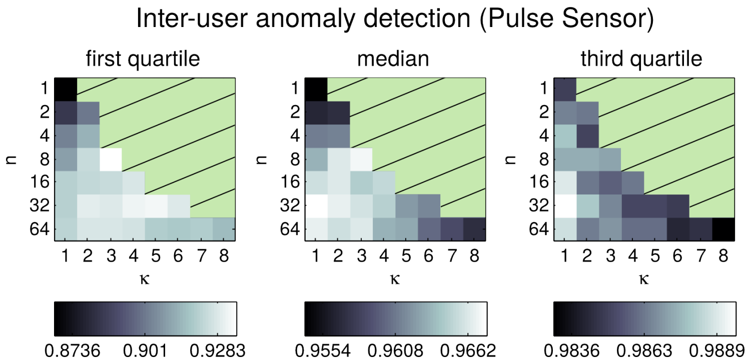

Fig. 4. Performance of several configurations of the considered algorithm in the inter-user anomaly detection. The three figures report the first quartile, the median and the third quartile of the AUC values computed on the Pulse dataset. The best configuration corresponds to $n = 8$, and $\kappa = 3$, as the performance degrades when considering simpler models (small n, κ) and more flexible ones (large n, κ). The intensity ranges in the three images are different for visualization sake.

are acquired from a different user or from a different device position. We use the KSVD algorithm [24] to learn a dictionary D from each of these 20 ECG tracing, using 500 randomly selected heartbeats[5]. Thus, for each dictionary D we consider normal those heartbeats belonging to the same tracing used to learn D (namely the same pair user-position), and anomalous those heartbeats from any different tracing.

We test the following number of atoms $n \in \{1, 2, 4, 8, 16, 32, 64\}$ in D and levels of sparsity $\kappa \in \{1, 2, 3, \ldots, \lceil n^{1/2} \rceil\}$. These settings are quite different from those traditionally used in image and signal processing, where $n > p$, yielding redundant dictionaries. However, we experienced heartbeats can be properly described by fewer atoms.

Figure 4 shows the performance on *inter-user* anomalies, where the anomalous heartbeats come from different users. More precisely, we report the three quartiles of the AUC values computed over the $20 \cdot 18 = 360$ combinations of ECG tracings from different users. Overall, the AUC values are quite large and this indicates that the considered algorithm can effectively discriminate between users. The *best* performance are achieved when $n = 8$ and $\kappa = 3$. Observe that the *single-atom* configuration ($n = 1$ and $\kappa = 1$) which reconstructs heartbeat by scaling a single atom to match at best the heartbeats, achieves significantly lower performance, as confirmed by a Wilcoxon signed-rank test (p-value $\approx 10^{-16}$).

Figure 5(a) shows the ROC curves on *intra-user* anomalies, where we consider as anomalous heartbeats acquired from the same user but with the device in a different position. These curves are averaged over all the possible 20 combinations of the ECG tracings, and we report only the *best* and *single-atom* con-

[5] We have observed that larger training sets do not lead to an improvement in the anomaly-detection performance.

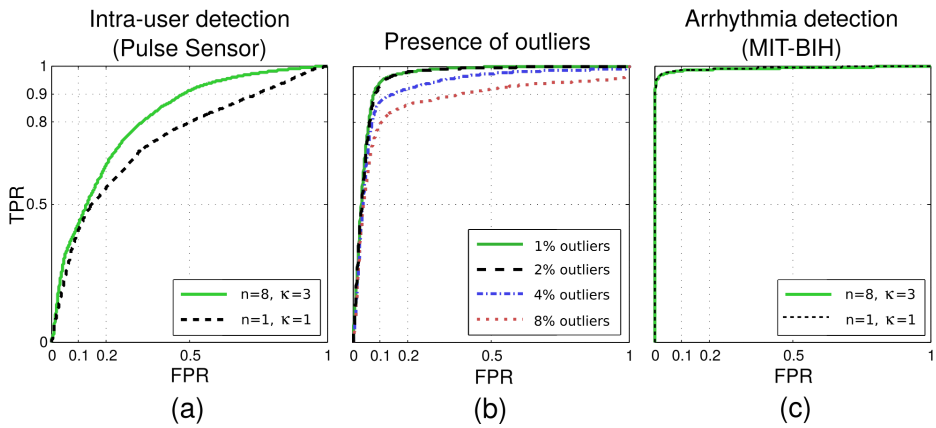

Fig. 5. ROC curves computed on Pulse dataset (a), (b), and on MIT-BIH Arrhythmia Database (c). In (a) two different configurations of parameters n and κ are considered in the intra-user anomaly detection. The best configuration clearly outperforms the single-atom one, confirming that a too simple model can not properly represent the structure of normal heartbeats. In (b) we consider the inter-user anomaly detection problem when the training set is corrupted by different percentages of outliers. This algorithm can tolerate small percentages of outliers, as its performance clearly degrades when the outliers reach 8% of training data. In (c) we compare the *best* and *single-atom* configuration in the arrhythmia detection problem. The Wilcoxon signed-rank test reveals no statistical evidence between the performance of the two configurations (p-value = 0.13), and both achieve very high performance.

figurations. Still, changes in the device positioning can be better detected when using multiple atoms than a single one. The AUC values are typically lower than in the inter-user case (the median AUC here 0.81 and 0.77 in the best and single-atom settings, respectively), and this indicates that in this dataset, intra-user differences are more subtle than inter-user differences.

These experiments confirm that it is necessary to use a quite flexible model to properly characterize the variety of normal heartbeats acquired by the Pulse Sensor and that dictionaries yielding sparse representations can successfully learn the heartbeat morphology of each user.

Finally, we remark that in ECG tracings acquired from wearable devices, user's movements can introduce low quality heartbeats, i.e., outliers, that might impair dictionary learning. Thus, we repeat the inter-user anomaly-detection experiment to assess whether the considered algorithm can tolerate small percentage of outliers in the training data. In particular, we consider the best configuration and introduce in the training sets of 500 heartbeats, $1\%, 2\%, 4\%, 8\%$ of outliers, which are selected among those heartbeats that were initially discarded. This experiment is repeated 15 times, and the average ROC curves are reported in Figure 5(b). It can be seen that the performance of the anomaly detection are stable when including only 1% and 2% of outliers, but dramatically decreases when outliers are 8%. This suggests that it is necessary to reduce the number of

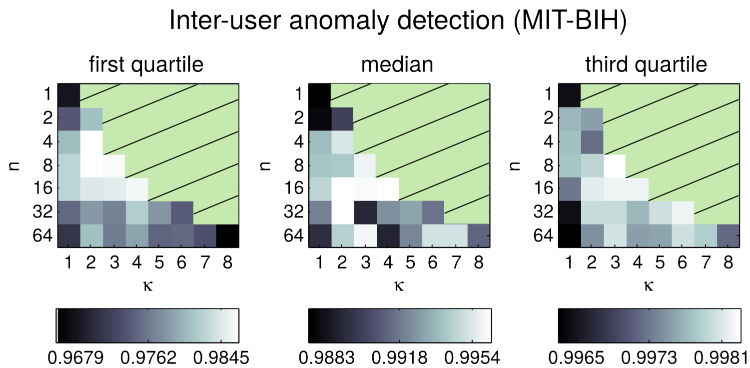

Fig. 6. Performance of several configurations of the considered algorithm for inter-user anomaly detection. The three figures reports the first quartile, the median and the third quartile of the AUC values computed on the MIT-BIH Arrhythmia Database. The best configuration corresponds to $n = 8$, $\kappa = 3$, as for the experiment in Figure 4. The intensity ranges in the three images are different for visualization sake.

outliers from the training set, e.g., by some prescreening method that analyzes MEMS recordings that are embedded on the Pulse Sensor.

5.4 Experiments on MIT-BIH Arrhythmia Database

We design two experiments also on the MIT-BIH Arrhythmia Database. In the first one, we show that our method can successfully detect *inter-users* changes also in this dataset, and that the performance are higher than in the Pulse dataset. As in the previous experiment we learn a dictionary D from 500 normal heartbeats of each tracing, considering the same range of parameters as in the Pulse dataset. AUC values are reported in Figure 6 and indicate that the best settings are the same ($\kappa = 3$ and $n = 8$). The Wilcoxon signed-rank test confirms that these parameters yield significantly superior performance than the single-atom settings (p-value $\approx 10^{-16}$). However, in all these settings, the median AUC is very close to 1, indicating very good detection performance independently of the parameters adopted. This suggests that the ECG tracings in the Pulse dataset are more difficult to model than in the MIT-BIH Arrhythmia Database, and we speculate that this is due to the fact that the heartbeats from MIT-BIH Arrhythmia Database present a low variability than in the Pulse dataset.

Finally, we assess the performance of the considered algorithm in an arrhythmia-detection task, using the annotations provided in the MIT-BIH Arrhythmia Database. In particular, we consider as anomalous the arrhythmias from the same patient used for dictionary learning. The ROC curves averaged over the entire dataset, for the *best* and *single-atom* configuration are very similar, and are reported in Figure 5(c). The Wilcoxon signed-rank test on the corresponding AUC values confirms that there is not a clear statistical evidence to claim that one configuration is better than the other (p-value $= 0.13$). This results can be explained by the fact that the arrhythmias show a very different morphology with respect to normal heartbeats, which allows these two methods to perform equally good.

6 Feasibility on the Pulse Sensor

We now investigate the overall feasibility of the considered anomaly-detection solution on the Pulse Sensor. In particular, we study both the requirements of dictionary learning, which is conveniently performed at each device positioning on an external host, and the computational complexity of the sparse coding, which has to executed in real time on the Pulse Sensor.

6.1 Dictoinary Learning and Device Configuration

Figures 4 and 5 confirm the need of learning the dictionary D every time the device is positioned, and at the same time indicate that 500 heartbeats are enough for this purpose. At an average heart rate, 500 heartbeats correspond to 7 minutes of ECG tracings, which can be conveniently transmitted via Bluetooth to an external host, e.g., the user smartphone, where the KSVD algorithm [24] can be executed[6] to learn the dictionary D, which is then sent back to the Pulse Sensor.

In the Pulse dataset these 7 minutes for training were acquired from users that were typically working in their office, thus performing normal actions and movements, while not in a rest state. In particular, experiments with outliers in the training set indicate that the dictionary learning in our specific settings (i.e., $n = 8$, $\kappa = 3$), can well tolerate a small percentage of heartbeats affected by user movements. Whenever higher robustness is requested, it is possible to leverage robust dictionary-learning algorithms that adopt an ℓ^1 norm for the data-fidelity term in (3), as in [28]. Alternatively, some form of pre-screening of the training set could be performed analyzing the MEMS recordings.

Let us finally remark that even if the device would provide sufficient computing power and memory for running the KSVD algorithm, it is nevertheless convenient to keep track of the training sets and learned dictionaries on an external host. It is in fact desirable to assess the quality of the recent acquisitions, for instance, by testing them with dictionaries previously learned.

6.2 Anomaly Detection on the Pulse Sensor

The ECG preprocessing we performed is the same as in [7], which consists in two median and a low-pass, convolutional, filter. Heartbeats are then segmented by locating the R-peaks using the Pan-Tompkins algorithm [29] and extracting a suitable temporal window centered in the R-peaks as in Section 5.1. All these operations are definitively compliant with computational capabilities of the Pulse Sensor.

[6] In the considered settings, the KSVD algorithm takes only few seconds on an ordinary laptop.

Anomalous heartbeats are then detected by solving the sparse coding problem (4), which represents the most time-demanding operation to be executed on the device. For this task, we adopt the OMP algorithm [25]: other anomaly-detection solutions based on sparse representations, like those in [19] and [27], are way more computationally demanding and cannot be implemented on the Pulse Sensor. The OMP is a greedy algorithm that solves (4) iteratively. In what follows we briefly illustrate the main steps of the OMP in its efficient implementation described in [30] (the same we used in our experiments), and we describe its computational complexity in terms of floating point operations (flop). At the very beginning $\mathbf{z} = D^T\mathbf{s}$ is computed at the cost of $\leqslant 2pn$ flop and the residual vector is defined as $\mathbf{r}^{(0)} = \mathbf{s}$. Then, the OMP iterates at most κ times the following steps, where l is used as an iteration index:

Correlation compute the inner product of the residual with each atom, i.e., $\mathbf{d}_k^T\mathbf{r}^{(l-1)}$, $k = 1, \ldots, n$ with an overall cost of $2pn$ flop.

Maximum select the atoms that is most correlated with $\mathbf{r}^{(l-1)}$, thus maximizing $|\mathbf{d}_k^T\mathbf{r}^{(l-1)}|$, $k = 1, \ldots, n$ which costs $\leqslant 2n$ flop.

Projection compute the coefficients $\mathbf{x}^{(l)}$ by orthogonal projection of $\mathbf{s}$ on the subspace spanned by the l atoms selected so far. This involves solving the linear system $\mathbf{z} = D_l^T D_l\mathbf{x}$, where D_l denotes the matrix containing all the selected atoms. Exploiting Cholesky factorization of $D_l^T D_l$, $\mathbf{x}^{(l)}$ can be computed at a cost $\leqslant 2pl + 3l^2$ flop.

Update update the residual $\mathbf{r}^{(l)} = \mathbf{s} - D\mathbf{x}^{(l)}$ at a cost of $\leqslant 2lp + p$ flop.

Considering the best parameters identified in our experiments (i.e., $p = 155, n = 8$, and $\kappa = 3$), a full execution of the OMP algorithm requires approximately 16K flop, which seems to be compliant with real-time operations on the Pulse Sensor. However, to make sure that the overall Pulse Sensor computing capabilities can guarantee real-time operation, we have performed some tests (with the same parameter values) directly on the device. In particular, thanks to the sensor suite modularity, we measured the execution times on the STM32F401 processor embedding a Cortex$^{\mathrm{TM}}$-M4F CPU with floating point unit (FPU). Tests were conducted using two versions of the CMSIS DSP library[7]: disabling/enabling the FPU optimization. The execution times when disabling the FPU optimization are a good estimate of the execution times on the STM32F103 processor that is actually used on the Pulse Sensor (which embeds a Cortex$^{\mathrm{TM}}$-M3 CPU without FPU). In this case, the OMP algorithm took 58.13ms allowing 17 executions per second at the maximum frequency of 72MHz, confirming the concrete possibility of executing the algorithm on this device within the period of an heartbeat. When enabling the optimization for FPU the OMP algorithm took only 6.58ms allowing 152 executions per second. These results are particularly encouraging since, realistically, a STM32F401 processor with FPU is going to be adopted in future embodiments of the Pulse Sensor.

[7] CMSIS DSP Software Library,
`https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html`

7 Conclusions

In this paper we investigate the problem of learning models to represent heartbeat morphology, in particular for monitoring ECG tracings acquired from wearable devices. Our study, conducted on ECG tracings form the Pulse Sensor, shows that dictionaries yielding sparse representations can effectively model the heterogeneous morphology of these heartbeats. In particular, we show that dictionaries can be successfully used to detect heartbeats having a morphology that is different from the training ones, and that this model can be effectively used in online monitoring schemes, implemented directly on the Pulse Sensor. Dictionary learning instead can be conveniently performed on an external host as it requires a limited amount of data to be transferred. Ongoing work concerns techniques to make the device configuration robust to user movements during the acquisition of the training set, which can be reasonably performed by pre-screen outliers in the MEMS recordings.

References

1. "The Pulse Sensor Project." `http://www.mrd-institute.com/#!projects/c1iwz`.
2. B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, pp. 582–588, 1999.
3. R. Hoekema, G. J. Uijen, and A. Van Oosterom, "Geometrical aspects of the interindividual variability of multilead ecg recordings," *Biomedical Engineering, IEEE Trans. on*, vol. 48, no. 5, pp. 551–559, 2001.
4. Y. H. Hu, S. Palreddy, and W. J. Tompkins, "A patient-adaptable ecg beat classifier using a mixture of experts approach," *Biomedical Engineering, IEEE Trans. on*, vol. 44, no. 9, pp. 891–900, 1997.
5. P. de Chazal and R. B. Reilly, "A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features," *Biomedical Engineering, IEEE Trans. on*, vol. 53, no. 12, pp. 2535–2543, 2006.
6. J. Wiens and J. V. Guttag, "Active learning applied to patient-adaptive heartbeat classification," in *Advances in neural information processing systems*, pp. 2442–2450, 2010.
7. P. De Chazal, M. O. Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *Biomedical Engineering, IEEE Trans. on*, vol. 51, no. 7, pp. 1196–1206, 2004.
8. N. P. Hughes, L. Tarassenko, and S. J. Roberts, "Markov models for automated ecg interval analysis.," in *NIPS*, pp. 611–618, 2003.
9. F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *Information Technology in Biomedicine, IEEE Trans. on*, vol. 12, no. 5, pp. 667–677, 2008.
10. S. Osowski, L. T. Hoai, and T. Markiewicz, "Support vector machine-based expert system for reliable heartbeat recognition," *Biomedical Engineering, IEEE Trans. on*, vol. 51, no. 4, pp. 582–589, 2004.
11. F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig, "Principal component analysis in ecg signal processing," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 98–98, 2007.

12. S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ecg classification by 1d convolutional neural networks.," *IEEE Trans. on bio-medical engineering*, 2015.
13. Z. Syed, J. Guttag, and C. Stultz, "Clustering and symbolic analysis of cardiovascular signals: discovery and visualization of medically relevant patterns in long-term data using limited prior knowledge," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 97–97, 2007.
14. H. Aidos, A. Lourenço, D. Batista, S. R. Bulò, and A. Fred, "Semi-supervised consensus clustering for ecg pathology classification," in *Machine Learning and Knowledge Discovery in Databases*, pp. 150–164, Springer, 2015.
15. G. B. Moody, "Mit-bih arrhythmia database directory." `https://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm`, May 1997.
16. J. L. Willems, P. F. Poblete, and H. V. Pipberger, "Day-to-day variation of the normal orthogonal electrocardiogram and vectorcardiogram," *Circulation*, vol. 45, no. 5, pp. 1057–1064, 1972.
17. A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.
18. J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *arXiv preprint arXiv:1411.3230*, 2014.
19. A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," *Journal of Signal Processing Systems*, vol. 79, no. 2, pp. 179–188, 2015.
20. B. Mailhé, R. Gribonval, F. Bimbot, M. Lemay, P. Vandergheynst, and J.-M. Vesin, "Dictionary learning for the sparse modelling of atrial fibrillation in ecg signals," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 465–468, IEEE, 2009.
21. J. Wang, M. She, S. Nahavandi, and A. Kouzani, "Human identification from ecg signals via sparse representation of local segments," *IEEE Signal Processing Letters*, vol. 20, pp. 937–940, Oct 2013.
22. C. Alippi, G. Boracchi, and B. Wohlberg, "Change detection in streams of signals with sparse representations," in *ICASSP*, pp. 5252–5256, IEEE, 2014.
23. M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.
24. M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
25. J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, pp. 2231–2242, Oct 2004.
26. S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM*, vol. 20, no. 1, pp. 33–61, 1998.
27. G. Boracchi, D. Carrera, and B. Wohlberg, "Novelty detection in images by sparse representations," in *IES*, pp. 47–54, IEEE, 2014.
28. C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proceedings of CVPR*, pp. 415–422, 2013.
29. J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *Biomedical Engineering, IEEE Trans. on*, no. 3, pp. 230–236, 1985.
30. R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.