A deep learning approach for ECG-based heartbeat classification for arrhythmia detection

G. Sannino *, G. De Pietro

National Research Council of Italy (CNR), Institute for High-Performance Computing and Networking (ICAR), Via P. Castellino 111, Naples, Italy

HIGHLIGHTS

- We present a Deep Neural Network (DNN) for the heartbeat classification.
- The DNN has been developed by using Tensor Flow and Google deep learning library.
- The proposed DNN is composed of 7 hidden layers and has been built empirically.
- Experiments have been carried out to test the proposed deep learning approach.
- Numerical results have shown the effectiveness of the approach.

ARTICLE INFO

ABSTRACT

Classification is one of the most popular topics in healthcare and bioinformatics, especially in relation to arrhythmia detection. Arrhythmias are irregularities in the rate or rhythm of the heartbeat which, in some cases, may occur sporadically in a subject's daily life. To capture these infrequent events, a Holter device is usually employed to record long-term ECG data. Therefore, the automatic recognition of abnormal heartbeats from a large amount of ECG data is an important and essential task. In the last two decades, a huge number of methods have been proposed to address the problem of ECG beat classification. At the same time, deep learning has advanced rapidly since the early 2000s and now demonstrates a state-of-the-art performance in various fields. In this paper, we propose a novel deep learning approach for ECG beat classification. We have conducted the experiments on the well-known MIT–BIH Arrhythmia Database, and compared our results with the scientific literature. The final results show that our model is not only more efficient than the state of the art in terms of accuracy, but also competitive in terms of sensitivity and specificity.

1. Introduction

The electrocardiogram (ECG) is a non-invasive, inexpensive and well established diagnostic tool widely used in several applications [1–3]. It represents the changes of the electrical activity of the heart over time and contains essential physiological information that is widely used to analyze heart function.

The cardiac electric field is recorded by means of electrodes on the body surface. It is measured as a voltage (i.e. potential difference) between two electrodes. A combination of two electrodes that forms an imaginary line in the body along which the electric signals are measured is called a "lead".

ECG signals are periodic signals, because they are composed of a sequence of waves that repeat periodically in time: a P wave, then Q, R and S waves (which form the QRS complex) and a T wave, as shown in Fig. 1. Very rarely, can a U wave also be detected. The most characteristic part of an ECG signal is the QRS complex [4]. Specifically, the analysis of this complex corresponds to the beat-to-beat classification.

An arrhythmia is an irregularity in the rate or rhythm of the heartbeat. During an arrhythmia, the heart can beat too fast, too slow, or with an irregular rhythm. Most arrhythmias are harmless, but some can be serious or even life-threatening. During an arrhythmia, the heart may not be able to pump enough blood to the body. Such a lack of blood flow can damage the brain, heart and other organs.

At critical levels, cardiac arrhythmias can be categorized into two types, life-threatening and non-life-threatening. Life-
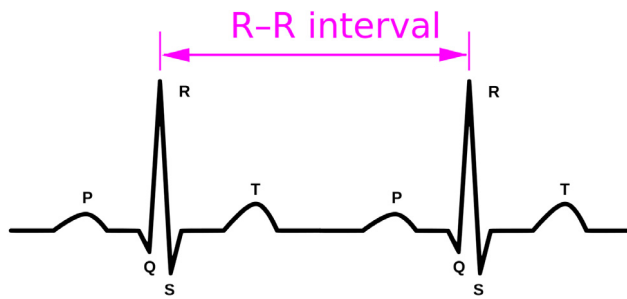
* Corresponding author.
E-mail addresses: giovanna.sannino@icar.cnr.it (G. Sannino), giuseppe.depietro@icar.cnr.it (G. De Pietro).

Fig. 1. A typical ECG trace, with a P wave, Q, R and S waves (which form the QRS complex) and a T wave, and the R-R interval labeled.

threatening arrhythmias, such as ventricular fibrillation and tachycardia, may trigger cardiac arrest and sudden death. Patients in these situations require emergency treatment. While non-life-threatening arrhythmias may not lead to imminent heart failure, timely therapy is still needed to avoid any further deterioration of the heart function. In some cases, arrhythmias may only occur sporadically in a patient's daily life.

To capture these infrequent events, a Holter device is usually employed to record long-term ECG data. Therefore, the automatic recognition of abnormal heartbeats from a large amount of ECG data is an important and essential task. The first type of arrhythmia has been well studied and some detection algorithms have been developed and implemented in automatic external defibrillators (AED) [5–7]. To recognize the second type, detecting the ectopic beat is a critical step.

Furthermore, for the analysis of long-term ECG recordings, beat-by-beat manual examination is tedious and time-consuming, especially in the case of bedside monitoring or wearable health care monitoring, where real-time diagnosis is a difficult task for junior doctors. Therefore, clinicians usually employ computer-assisted methods to analyze and interpret the ECG signals.

As also remarked in [8], thanks to the rapid growth of novel sensing technologies, the ECG data to be analyzed is not only enormous in amount but also complex in structure for traditional software and hardware. In fact, some reports have estimated that digital healthcare data from all over the world totaled 500 petabytes (1015) in 2012. Moreover, it is expected to increase to reach 25 exabytes in 2020 [9].

Hence, solutions are needed in order to manage and analyze such complex, diverse, and huge datasets with a reasonable time complexity and storage capacity. Big data analytic, a popular term given to datasets that are large and complex, plays a vital role in managing the huge amount of healthcare data and improving the quality of healthcare services offered to patients. In this context, one of the challenges lies in the classification of this data, which relies on effectively distributed processing platforms and advanced data mining and machine learning techniques.

Therefore, a deep learning technique is introduced in this work to meet the challenges faced by classify the ECG beats. Recently, deep learning techniques have been used by many companies, including Adobe, Apple, Baidu, Facebook, Google, IBM, Microsoft, NEC, Netflix, and NVIDIA [10], and in a very large set of application domains, as for example in [11,12]. In this paper, we present a novel deep learning approach for ECG beat classification. The approach has been developed by using the Tensor Flow framework, the deep learning library from Google, in the Python programming language.

We have conducted experiments on the well-known MIT–BIH Arrhythmia Database, and compared our results with the scientific literature. The final results show that our model is not only more efficient than the state of the art in terms of accuracy, but also competitive in terms of sensitivity and specificity.

In the following section, Section 2, we report an overview of the existing automatic heartbeat classification techniques and algorithms, and comment on their features and characteristics. The ECG-based heartbeat classification model is presented in Section 3, with a detailed description of the MIT–BIH Arrhythmia Database (MIT–BIH-AR) provided in the Section 3.1. Details about the signal processing used to create the new dataset are given in Section 3.2, while a description of its characteristics is provided in Section 3.3. The proposed Deep Neural Network (DNN) for the heartbeat classification is described in Section 3.4, and its performance in comparison with other well-known classifiers is shown in Section 4 in terms of accuracy, sensitivity and specificity. Finally, our conclusions and ideas for future work are presented in Section 5.

2. State of the art

A considerable amount of research has been dedicated to automatic heartbeat classification [13–16]. The works can be categorized into two classification paradigms, namely, "intra-patient" and "inter-patient", also known as "class-oriented" and "subject-oriented" respectively [17]. The intra-patient paradigm [13,14] partitions the whole dataset into training and testing subsets based only on the beat label, and therefore an ECG recording may partly appear in both the data subsets. With this scheme, the classifiers usually produce over-optimistic results [18]. In clinical practice, the classification performance declines due to the inter-individual variation.

In order to adapt the dataset to practical situations, de Chazal et al. [15] proposed the inter-patient paradigm where the training and testing subsets were constructed from different ECG recordings so that the inter-individual variation would be taken into account and the classifier would exhibit a better generalization ability. In [15] the best-performing single-lead classifier obtained a maximum accuracy of 83.0%. Instead, the sensitivity and specificity results reveal that, while it achieved a high specificity (88.1%), the resulting sensitivities were very low.

The inter-patient paradigm has been adopted also in [16,17,19] to evaluate classification performance in the MIT–BIH-AR dataset [20,21].

In addition, a hybrid paradigm called "patient-specific" was also proposed by de Chazal et al. [22], in which, first, a global classifier was trained and then a local classifier was employed to tune the global classifier. The gross performance of the system presented in [22] was accuracy 97.4%, sensitivity 94.4%, and specificity 98.4%. It was reported that this approach outperformed those achieved by pure inter-patient classifiers [7]. Inspired by this paradigm, Ince et al. [23] proposed a "patient-specific" neural network (NN) where the training data consisted of two parts: (i) common representative beats randomly selected from the training recordings and (ii) patient-specific beats segmented from the first 5 min of each recording. Wiens et al. [24] employed a Supervised Vector Machine (SVM) active learning method for heartbeat classification, instead, Llamedo et al. [16] studied how much the performance would be improved when experts efficiently assisted in the active learning iteration steps.

Most of the studies followed the Advancement of Medical Instrumentation (AAMI) recommendation [25] which specifies heartbeats using five labels, namely normal or bundle branch block beat (N), supra-ventricular ectopic beat (S), ventricular ectopic beat (V), fusion of ventricular and normal beat (F), and heartbeats that cannot be classified (Q). This recommendation makes it possible to perform an accurate comparison between various heartbeat classifiers. Regarding the ECG features commonly employed for

the classification, features surrounding the R-R intervals are those most widely used. The R-R interval is defined as the interval from the peak of one QRS complex to the peak of the next, as shown on an electrocardiogram in Fig. 1. The most commonly used R-R features are the pre-RR, post-RR, local RR, and average RR [15,16].

Other time domain features including the PP interval, P duration, QRS duration, PR interval, T duration and QT interval are also considered. Moreover, "morphology" features of the ECG samples in the P wave, QRS complex and T wave, as well as the morphological distances [24] between the beats and the median beat, have also been used. These features have been clinically studied and the related diagnostic standards have also been stipulated.

Although vectorcardiogram (VCG) based features [16,26] can provide comprehensive information about the heart condition, the reconstruction of the VCG requires more leads and thus the applicability of these features is rather limited.

A fully automatic configuration (GE classifier) approach presented in [27] achieved an accuracy of 75.3%, a sensitivity of 69.6%, and a specificity of 76.6%. These results were obtained by using a semi-automatic mixture of experts (MOE) approach.

Additionally, frequency domain feature analysis can also provide considerable insight into ECG signals. Signal processing methods include wavelet decomposition 100 (WT) [13,14,28], principal component analysis (PCA) and independent component analysis (ICA) [17]. Although these features are associated with clear mathematical interpretations, they do not have a physiological meaning that allows doctors to comprehend them in an intuitive way. Hence, the selection of the most relevant features from among the mass of potential features to improve the classification performance is a great challenge [29].

Llamedo et al. [16] employed a Sequential Floating Feature Selection (SFFS) algorithm to enhance the Bayesian classifier. They tried to find an optimal feature subset for all the sub-classifiers. However, in clinical practice, different diseases are characterized by different features (symptoms). By analyzing these features, clinicians are able to differentiate one disease from another, or distinguish a diseased state from a healthy state.

Based on these considerations, we have attempted to develop a novel deep learning based approach to enhance the performance of heartbeat classification by using some samples of the beats and the features surrounding the RR intervals. To evaluate this approach, we have performed some comparisons with other well-known classifiers in order to ensure a better generalization ability in the differentiation of abnormal beats from normal ones.

3. The ECG-based heartbeat classification model

3.1. The MIT–BIH Arrhythmia Database

In this work, we have used the MIT–BIH Arrhythmia Database [20], a dataset of standard test material used since 1980 in innumerable scientific works for the evaluation of arrhythmia detectors and classifiers. It was compiled by collecting 24-hour ECG recordings from 47 subjects, specifically 25 men aged 32 to 89 years and 22 women aged 23 to 89 years.

The acquisition of the ECG recordings was achieved by using Del Mar Avionics model 445 two-channel reel-to-reel Holter recorders, and each signal was then digitized by using a Del Mar Avionics model 660 playback unit. The hardware used for the digitization was designed and built in the BIH Arrhythmia Laboratory, including the tape-drive controllers and the analog-to-digital converter (ADC) interfaces. To provide for the use of simple digital notch filters, able to remove the main frequency interference at 60 Hz, a digitization rate of 360 samples per second per channel was chosen.

Each ECG record of the MIT–BIH Arrhythmia Database includes two leads originating from different electrodes. The most common leads in the database are the modified limb lead II (MLII), and V1, obtained by placing the electrodes on the chest. An example of a record extracted from the MIT–BIH database is reported in Fig. 2.

A subset of the recordings, the "200 series", contains uncommon but clinically important arrhythmias that would not be well represented in a small random sample (see Fig. 3).

3.2. Signal processing

Starting from the MIT–BIH Arrhythmia Database we have created a new dataset by applying to the ECG signals the following processing steps, summarized in Fig. 4:

- **Denoising**: to eliminate possible power line interference and baseline wanderings caused by respiration or patient movements that can create problems in the detection of the ECG peaks.
- **Peak Detection**: to determine the positions of all peaks of interest found in the ECG signal. The p-wave and t-wave positions are useful for the signal segmentation. Instead, the r-peak detection is essential for the temporal features extraction.
- **Signal Segmentation**: to divide the ECG signal into single beats that will be classified as normal or abnormal.
- **Temporal features extraction**: to extract information about the ECG signal trend or, more precisely, about the variability of the signal.

All these processing steps were carried out on the V1 lead of each subject of the MIT–BIH Arrhythmia Database by using Matlab [30], version R2014b (8.4.0.150421), distributed by Mathworks, launched on an iMac with the macOS Sierra version 10.12.6.

3.2.1. Denoising

As discussed in [31,32], the ECG signal is normally corrupted by various kinds of noise signal, namely artifacts, e.g. power line noise, of which the most common are well highlighted in [33]. Due to these artifacts, the information of interest cannot be readily extracted from the raw signal, which instead must be processed to model and to denoise the ECG signal.

A noisy ECG signal extracted from the MIT–BIH database is shown in Fig. 5.

Among the several sources of noise that can corrupt the ECG signal, and consequently lead to a wrong diagnosis and/or classification, the most common sources are *power line interference* [33], and *baseline wandering*. Power line interference consists of 60/50 Hz pick-up and harmonics and the amplitude is 50% of the peak-to-peak ECG amplitude. Some of the common causes of the 50 Hz interference are reported in [34]. Instead, baseline wandering is caused by respiration or patient movement which creates problems in the detection of the ECG peaks.

Even if in the scientific literature there are numerous manuscripts dedicated to the description of innovative and high-performance algorithms for the denoising of the ECG signal, as for example [32,35], we chose to apply the same denoising techniques as other studies [15,22,36] addressed at the same aim of classification. This decision was taken to make possible a correct comparison of the proposed DNN with the literature. Therefore, we denoised the ECG signals by using: (i) two median filters and (ii) a 12-order low pass filter with a 35 Hz cut-off frequency.

Applying a median filter of length n to the ECG signal x, means that a sliding window of length n is applied to the ECG signal, and for each step the median value in the window is returned. $y(i)$ will be the median of:

$$\begin{cases} x(i-(n-1)/2) : x(i+(n-1)/2) & \text{if } n \text{ is odd,} \\ x(i-n/2), x(i-(n/2)+1), \ldots, x(i+(n/2)-1) & \text{if } n \text{ is even.} \end{cases}$$
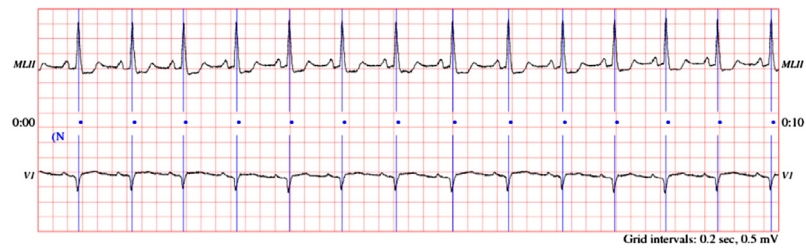
$$(1)$$

Fig. 2. ECG traces of the two main derivations (leads), namely MLII and V1, for a normal subject. Vertical lines represent the position of the R-peak.
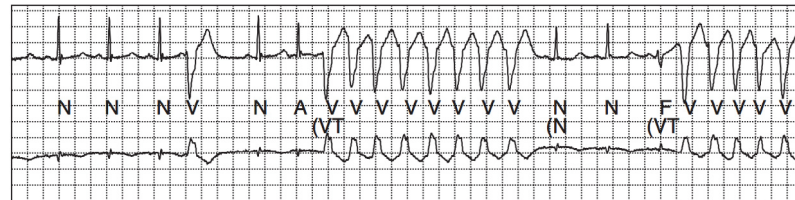*Source:* Figure taken from Physionet.org.

Fig. 3. Ten seconds from record 205 of the MIT–BIH Arrhythmia Database. Rigorously reviewed beat annotations (A: atrial premature beat, F: ventricular fusion beat, N: normal beat, and V: ventricular premature beat) and rhythm annotations ((N) normal sinus rhythm and (VT) ventricular tachycardia) appear in the center, between the two ECG signals: above MLII, and below V1).
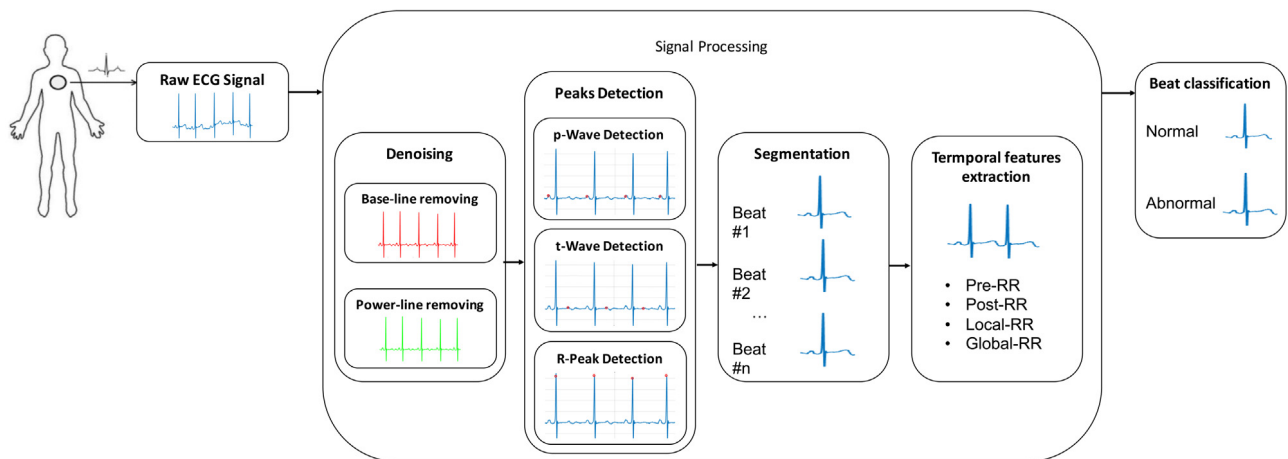*Source:* Image taken from [21].

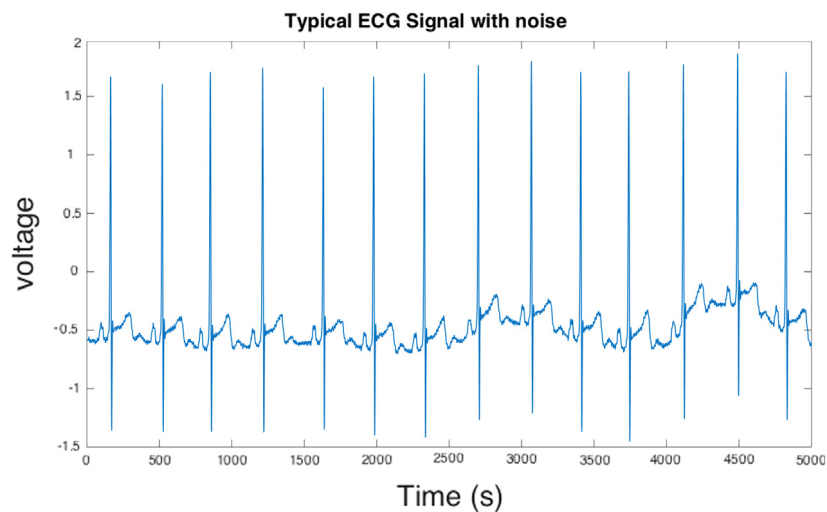**Fig. 4.** Overview of all processing steps.

Fig. 5. Typical ECG signal with noise: subject number 115 of the MIT–BIH database.
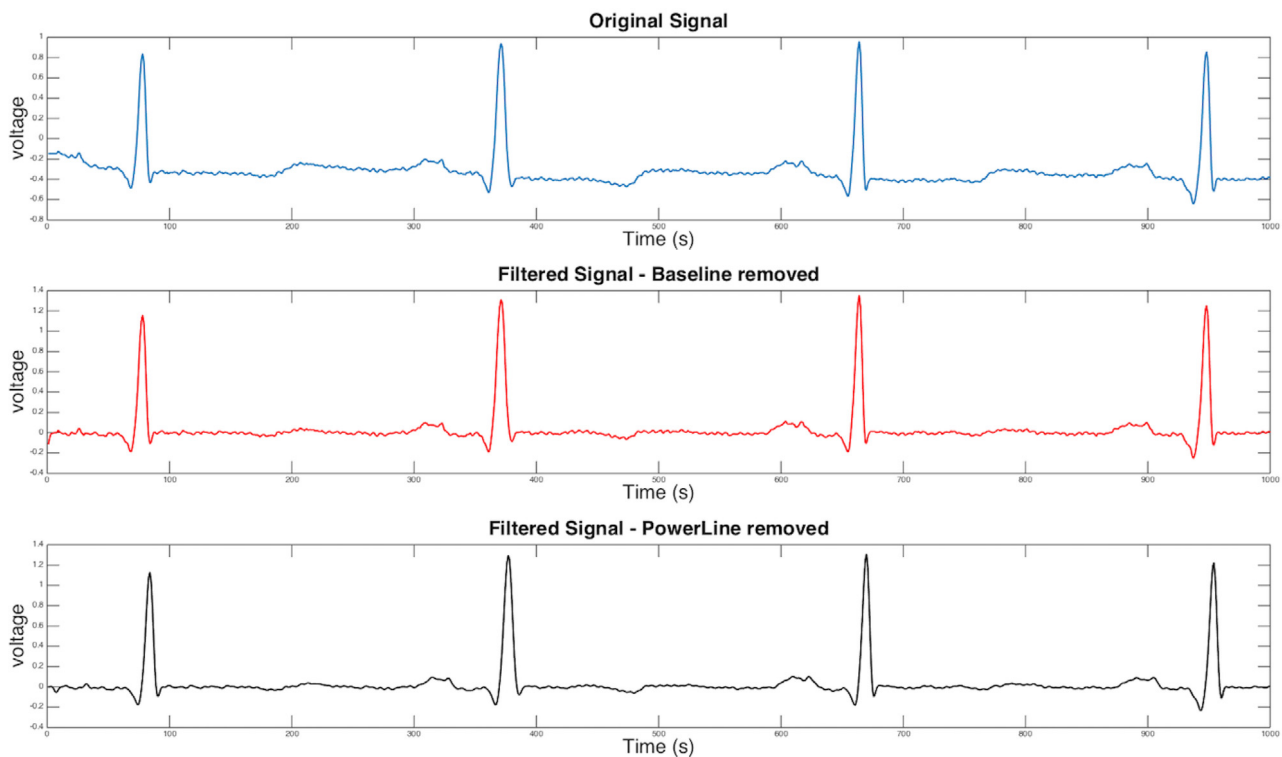
Fig. 6. Denoising steps: the raw ECG signal is reported in blue (the original signal of subject 100 of the MIT-BIH database), potentially affected by power line and baseline interference; the filtered ECG signal, after the application of the two median filters able to remove the baseline wandering, is reported in red; finally, the filtered ECG signal, after the application of the two median filters and the $n = 12$ FIR filter able to remove also the power line interference, is reported in black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To apply two median filters, one with a sliding window of length $n = 600$ ms and another with $n = 200$ ms, we used the Matlab function medfilt1 [37] that implements one-dimensional median filtering. The advantage of the median filter is that it neatly removes outliers while adding no phase distortion. In contrast, the regular infinite impulse response (IIR) and finite impulse response (FIR) filters are nowhere near as effective at removing outliers, even with high orders. The signal resulting after the two median filters had been applied is reported in red in Fig. 6. It is possible to note how the filters eliminate the oscillatory trend of the signal caused by the baseline interference.

After the application of the two median filters, we applied an order $n = 12$ FIR filter with the given frequency cut-off $w = 35$ Hz, by using the Matlab function fir1 [38]. This function implements the classic method of the windowed linear phase FIR digital filter and provides filters in standard low pass, band pass, high pass, and band pass configurations. The filtered signal, obtained by applying the two median filters and the $n = 12$ FIR filter, is reported in black in Fig. 6. It is possible to note how the filters adjust the oscillation frequency.

3.2.2. Peak detection

After filtering the signals, we carried out the peak detection phase useful for the segmentation of the ECG signal into single beats. To achieve this aim, we chose to use the WaveForm DataBase (WFDB) Toolbox [39] available from Matlab, applying the function wqrs() to the ECG signal, which provided us with the positions of all peaks of interest found in the ECG signal.

Using this technique, the QRS detection, especially the detection of the R wave, is easier, in that it is more clearly distinguished from other parts of the ECG signal due to its structural form and high amplitude. Each R peak detection corresponds to the detection of

a single heartbeat. The QRS detector is based on the algorithm of Pan and Tompkins [40], with some improvements that make use of slope information [41].

This information, together with the QRS complexes and the ECG signal, provides the input to the WFDB function ecgpuwave(), which gives us the exact position of all the P, R, and T peaks found in the signal, as shown in Fig. 7. The output of the ecgpuwave is written as a standard WFDB-format annotation file associated with the specified annotator. The output annotation file contains PWAVE ("p") and TWAVE ("t") annotations that indicate the P- and T-wave peaks.

3.2.3. Signal segmentation

Having located the P, R, and T peaks, we proceeded to segment the ECG signal into single heartbeats, as shown in Fig. 8.

For each beat, we chose to take 50 uniformly distributed samples, in accordance with [15,22,36], starting from the p-peak until the t-peak detected in the previous step.

3.2.4. Temporal features extraction

Dividing the ECG signal into single beats causes a loss of information about the ECG waveform, specifically about the variability of the ECG signal.

In order to take into account also this information, which is useful for the classification of each beat, we computed four temporal features for each beat, in accordance with [15,22,36]:

- **the Pre-RR interval**, defined as the RR-interval between a given heartbeat and the previous heartbeat;
- **the Post-RR interval**, defined as the RR-interval between a given heartbeat and the following heartbeat;
- **the Local average RR interval**, defined as the average of the 10 RR-intervals within a sliding window covering the past 10 s; and
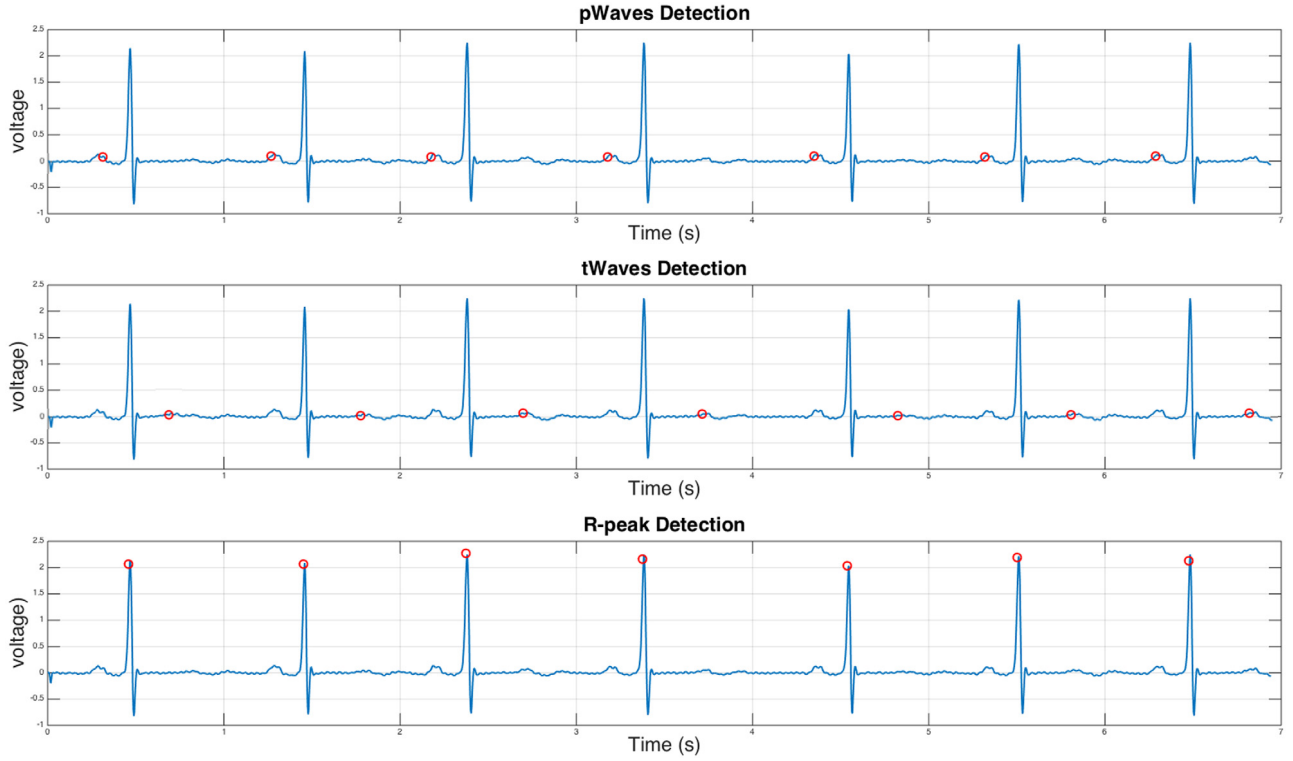
Fig. 7. Peak detection steps: at the top, the ECG signal of subject number 115 of the MIT–BIH database is reported, on which we detected the p-waves highlighted with red circles; in the middle, the same ECG signal is reported, on which we detected the t-waves highlighted with red circles; finally, at the bottom, the same ECG signal is reported, on which we detected the r-peaks highlighted with red circles. All the detections were made by using the WFDB function ecgpuwave().
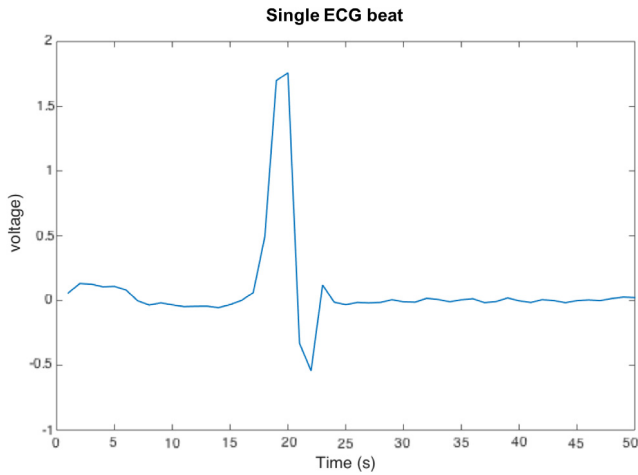

Fig. 8. An example of a single beat extracted from the ECG signal of subject number 115 of the MIT–BIH database.

- **the Global average RR interval**, defined as the average of the 10 RR-intervals within a sliding window covering the previous 5 min;

3.3. The dataset

For each subject of the MIT–BIH database, we computed the four pre-processing steps just described. Excluding the first five minutes of each record (subject), we obtained, considering the whole database, a total of 84,615 items. The justification for excluding the first five minutes for each subject lies in the impossibility of calculating the global average RR interval.

Each item i of the new dataset consists of:

$$i = sub_{id}; \ sample_n|_{n=1:50};$$
$$PreRR; \ PostRR; \ LocalRR; \ GlobalRR; \hspace{2cm} (2)$$
$$class$$

where:

- sub_{id} identifies the subject;
- $sample_1, \ldots, sample_{50}$ represent the beat samples;
- $PreRR$; $PostRR$; $LocalRR$; $GlobalRR$ are the extracted attributes defined in the previous section; and
- $class$ represents the class, encoded as 1 for normal beats and 2 for abnormal ones.

Of these 84,615 items, 66,750 represent normal beats (N), 2288 represent abnormal beats (premature ventricular contractions (V), supra-ventricular premature beats (S), or a fusion of ventricular and normal beats (F)), and 14,828 represent unclassifiable beats (Q).

Due to their non-classification, we removed from the dataset the last 14,828 items. Additionally, this was necessary in order to balance the dataset due to the fact that the classes were imbalanced, namely we had too many normal beats (N) compared to abnormal ones (V, S and F). In fact, in these cases, conventional algorithms are often biased towards the majority class because their loss functions attempt to optimize quantities such as error rate, not taking into consideration the data distribution. In the worst case, minority examples are treated as outliers of the majority class and are ignored, the learning algorithm simply generating a classifier that classifies every example as the majority class. To avoid these problems, we decided to select only 2288 items representing the normal beats class (N) from the initial 66,750, randomly selected from all the subjects.
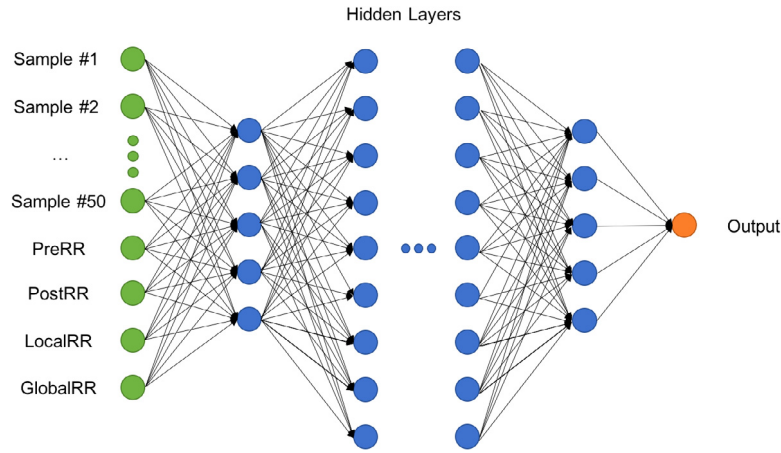
Hidden Layers

Fig. 9. An overview of the proposed DNN.

Table 1
Details of the dataset.

	Training set	Testing set	Whole dataset
Normal beats (N)	1466	822	2288
Abnormal beats (A)	1246	1042	2288
Total (# and %)	2712 (∼60%)	1864 (∼40%)	4576

Therefore, the final dataset was composed of a total of 4576 items, 2288 representing the normal beats class (N) and 2288 representing the abnormal beats class (A). This dataset, thanks to the information contained in sub_{id} in each item, was divided into the training (DS1) and test (DS2) set, in accordance with other similar works [15,22,36].

The two sets were composed of the following MIT–BIH-AR recordings:

- DS1 = {101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230},
- and DS2 = {100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234}.

More details about the item distribution over the testing and training sets are reported in Table 1.

3.4. Methodology: the proposed DNN

Artificial Neural Networks (ANN) are computational algorithms inspired by networks of biological neurons to solve prediction problems in computer vision, natural language processing and drug discovery, etc.

A DNN is a particular artificial neural network with more than 3 layers, which inherently fuses feature extraction and classification into a signal learning body and directly constructs a decision-making function. These kinds of neural networks have achieved great success in the complicated fields of pattern recognition in recent years.

Generally, a DNN consists of an input layer for the raw descriptors X_l, L hidden layers, and an output layer for the prediction. An overview of the proposed DNN is given in Fig. 9.

In detail, we have developed the DNN by using the Tensor-Flow framework [42], the `tf.contrib.learn.DNNClassifier` deep learning library from Google, in the Python programming language.

Currently, no standard methodology exists for the construction of an optimal neural network with the correct number of layers and number of neurons for each layer. For these reasons, we built our

DNN empirically, by performing a wide set of trials. In each trial, we have manually configured a DNN by changing the following parameters: the number of hidden layers, the activation function, the number of learning steps and, for each of the hidden layers, the number of neurons making up the layer.

For each manual configuration, we evaluated the classification accuracy over the testing set. After this laborious manual phase, we obtained the best classification performance with a DNN composed of seven hidden layers, with 5, 10, 30, 50, 30, 10 and 5 neurons, respectively.

The DNN Classifier class used here creates all the neuron layers, based on the ReLU (Rectified Linear Unit) activation function. Hence, observing the mathematical form in (3) of this function, we can see that it is very simple and efficient. The output layer relies on the softmax function, and the cost function is the cross entropy [43].

The rectifier is an activation function defined as:

$$f(x) = x^+ = \max(0, x) \tag{3}$$

where x is the input to a neuron. This is also known as a ramp function and is analogous to half-wave rectification in electrical engineering.

A unit employing the rectifier is called a Rectified Linear Unit (ReLU). A smooth approximation to the rectifier is the analytic function:

$$f(x) = \ln[1 + \exp(x)], \tag{4}$$

which is called the softplus function. The rectifier and softplus functions are shown in Fig. 10.

Upon prediction, a new representation of the raw descriptors was extracted from the hidden layers as follows:

$$X_{l+1} = H(W_l X_l + B_l), \qquad l = 1, \dots, L \tag{5}$$

where W_l and B_l are respectively the weight matrix and bias for the lth hidden layer, and H is the associated activation function, which is selected to be a rectified linear unit (ReLu).

A general pseudo-code describing the DNN is reported in the following:

- load the datasets, i.e. the training and testing sets;
- build the classifier by using
  `tf.contrib.learn.DNNClassifier` Google library according with the manual configuration selected, i.e. the number of hidden layers, the activation function, the number of learning steps, and, for each hidden layer, the number of neurons making up the layer;
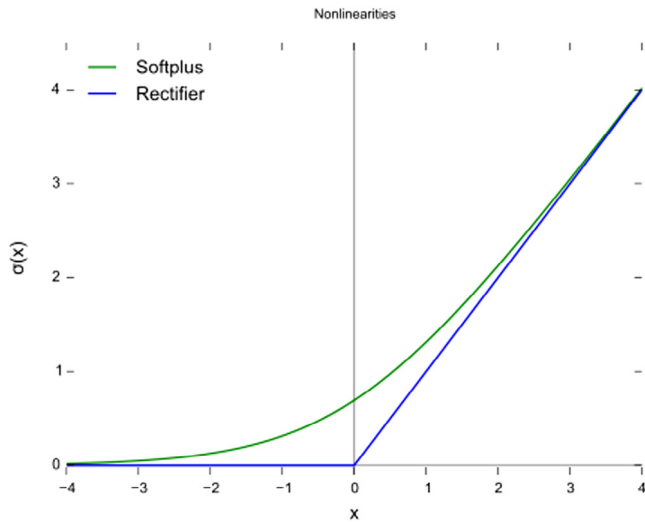
Fig. 10. Plot of the rectifier (blue) and softplus (green) functions near $x = 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- fit the model by using the `classifier.fit` function;
- evaluate the accuracy of the DNN over the training set by using the `classifier.evaluate` function;
- calculate the predictions of the DNN over the testing set by using the `classifier.predict` function;
- evaluate the classification performance of the DNN over the testing set by calculating the confusion matrix;
- calculate the classification performance of the DNN over the whole dataset.

4. Performance evaluation

For the evaluation of the proposed DNN, we assumed the class containing abnormal beats as the positive class, and therefore the accuracy, sensitivity, and specificity are computed with reference to this.

For the sake of clarity, we report the definitions for the above parameters.

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FP + FN)}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{FP + TN}$$

where:

- True Positive (TP) is the number of abnormal beats correctly classified;
- True Negative (TN) is the number of normal beats correctly classified;
- False Positive (FP) is the number of normal beats incorrectly classified as abnormal;
- and False Negative (FN) is the number of abnormal beats incorrectly classified as normal.

Table 2 reports the overall discrimination ability of the proposed DNN over all the considered datasets (the training, testing and whole datasets) in terms of accuracy, sensitivity, and specificity.

Table 2
Discriminating ability of the proposed DNN Classifier.

	Training set	Testing set	Whole data
Accuracy	100%	99,09%	99,68%
Sensitivity	100%	98,55%	99,48%
Specificity	100%	99,52%	99,83%

Table 3
Results for accuracy achieved by the tested classifiers.

	Training set	Testing set	Whole data
Naive Bayes [47]	91.76%	92.22%	91.92%
MLP [48]	99.54%	98.93%	99.32%
KStar [49]	**100%**	97.21%	99.02%
AdaBoostM1 [45]	97.08%	96.99%	97.05%
Bagging [46]	99.10%	98.87%	99.02%
OneR [50]	91.73%	90.87%	91.43%
Part [51]	99.80%	98.55%	99.36%
J48 [52]	99.77%	98.12%	99.19%
Random Tree [53]	**100%**	98.77%	99.57%
JRip [54]	99.57%	98.17%	99.08%
SVM [55–57]	97.08%	97.37%	97,18%
Our DNN	**100%**	**99.09%**	**99.68%**

Table 4
Results for sensitivity achieved by the tested classifiers.

	Training set	Testing set	Whole data
Naive Bayes [47]	86.61%	87.81%	87.04%
MLP [48]	99.86%	98.90%	99.52%
KStar [49]	**100%**	94.25%	97.86%
AdaBoostM1 [45]	98.10%	98.60%	99.28%
Bagging [46]	99.59%	99.38%	99.51%
OneR [50]	87.68%	87.80%	87.72%
Part [51]	99.93%	**99.13%**	**99.65%**
J48 [52]	99.86%	98.04%	99.21%
Random Tree [53]	**100%**	99.02%	99.65%
JRip [54]	99.73%	98.16%	99.17%
SVM [55–57]	95.96%	97.07%	96.36%
Our DNN	**100%**	98.55%	99.48%

To demonstrate the quality of the proposed DNN in terms of accuracy, sensitivity, and specificity, we carried out a precise comparison by taking into account eleven other well-known classifiers.

To achieve this aim, we used the most widespread tool in the scientific literature, the Waikato Environment for Knowledge Analysis (WEKA) system release 3.4 [44]. WEKA contains tools for classification, regression, data pre-processing and clustering etc. WEKA contains several classification techniques, separated into categories, such as for example lazy, meta-techniques, and Bayesian, and those based on functions, trees and rules.

For each category, we selected the most representative algorithms, in terms of classification performance. In detail, we considered AdaBoostM1 [45] and Bagging [46] from among the meta-techniques, Naive Bayes [47] from among the Bayesian techniques, and MultiLayer Perceptron Artificial Neural Network (MLP) [48] from among the function-based techniques. As representatives of the lazy methods, we selected KStar [49], while among the rule-based classification methods, we took into consideration OneR [50], and Part [51].

We chose J48 [52] and Random Tree [53] from among the tree-based classifiers, and, finally, we decided to consider the propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (JRip) [54] and the support vector machine classifier (SVM) [55–57] from among the other methods.

Tables 3–5 show in bold, for each parameter, the best value obtained by all the considered algorithms. These results highlight the fact that the accuracy of the proposed DNN over all the datasets (the training, testing and whole datasets) is the highest, with values greater than 99%.

Table 5
Results for specificity achieved by the tested classifiers.

	Training set	Testing set	Whole data
Naive Bayes [47]	96.26%	96.28%	96.27%
MLP [48]	99.30%	98.95%	99.18%
KStar [49]	**100%**	**99.80%**	**99.93%**
AdaBoostM1 [45]	96.37%	95.82%	96.18%
Bagging [46]	98.76%	98.48%	98.66%
OneR [50]	95.09%	93.51%	94.54%
Part [51]	99.70%	98.11%	99.15%
J48 [52]	99.70%	98.18%	99.18%
Random Tree [53]	**100%**	98.57%	99.51%
JRip [54]	99.45%	98.18%	99.02%
SVM [55–57]	97.92%	97.60%	97.81%
Our DNN	**100%**	99.52%	99.83%

Concerning sensitivity, the proposed DNN achieves a good value, inferior to the best, namely Part, by less than 0.2% on the whole dataset, and less than 0.06% on the testing set. This means that it has a lower number of false negatives, i.e. abnormal beats that are erroneously classified as normal, in comparison with the other algorithms.

Regarding specificity, kStar is the best. However, it is among the worst in terms of sensitivity, in that it reports serious problems in relation to false negatives. The proposed DNN is not the best, although it should be considered that it obtained a very good value, inferior to the best by only 0.10% on the whole dataset and by only 0.28% on the testing set, therefore meaning that it has a lower number of false positives.

5. Conclusions and future work

Classification is one of the most popular topics in healthcare and bioinformatics, especially for arrhythmia detection. Arrhythmias are irregularities in the rate or rhythm of the heartbeat which, in some cases, arrhythmias may occur sporadically in a subject's daily life. To capture these infrequent events, a Holter device is usually employed to record long-term ECG data. Therefore, the automatic recognition of abnormal heartbeats from a large amount of ECG data is an important and essential task.

In this paper, we have proposed an approach based on a Deep Neural Network (DNN) for the automatic classification of abnormal ECG beats, differentiated from normal ones. DNN has been developed by using the Tensor Flow framework, the deep learning library from Google, and it is composed of only seven hidden layers, with 5, 10, 30, 50 ,30, 10 and 5 neurons, respectively.

To demonstrate the quality, in terms of correct classification, of the proposed DNN, we have carried out a precise comparison by taking into account eleven other well-known classifiers thanks to the use of the WEKA tool. The numerical results have shown the effectiveness of the approach, especially in terms of accuracy.

Our future research will involve the use of other freely available databases, like INCART and SVDB, in order to evaluate the precision of the approach and of the proposed DNN, and also in order to optimize the learning ability of the neural network. Moreover, we are planning for our approach to be embedded in a real-time ECG monitoring system and to be tested in a real-world situation by means of a close co-operation with a specific hospital department.

Another very interesting issue worth investigating is that related to the involvement of physicians in the process, with reference to the annotations. In fact, as seen in the description of the database, there are currently too many ECG beats which have to be annotated for this to be used in the training or testing phase of the DNN.

Conflict of interest

The authors declare that the writing of this paper does not cause any competing interests to them.

References

[1] L. Carnevale, A. Celesti, M. Fazio, P. Bramanti, M. Villari, Heart disorder detection with menard algorithm on apache spark, in: European Conference on Service-Oriented and Cloud Computing, Springer, 2017, pp. 229–237.
[2] P. Melillo, R. Castaldo, G. Sannino, A. Orrico, G. De Pietro, L. Pecchia, Wearable technology and ECG processing for fall risk assessment, prevention and detection, in: 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, 2015, pp. 7740–7743.
[3] G. Sannino, G. De Pietro, An evolved ehealth monitoring system for a nuclear medicine department, in: Developments in E-systems Engineering (DeSE), IEEE, 2011, pp. 3–6.
[4] P. Laguna, R. Jané, S. Olmos, N.V. Thakor, H. Rix, P. Caminal, Adaptive estimation of QRS complex wave features of ECG signal by the Hermite model, Med. Biol. Eng. Comput. 34 (1) (1996) 58–68.
[5] A. Amann, R. Tratnig, K. Unterkofler, Detecting ventricular fibrillation by time-delay methods, IEEE Trans. Biomed. Eng. 54 (1) (2007) 174–177.
[6] U. Irusta, J. Ruiz, S.R. de Gauna, T. Eftestøl, J. Kramer-Johansen, A least mean-square filter for the estimation of the cardiopulmonary resuscitation artifact based on the frequency of the compressions, IEEE Trans. Biomed. Eng. 56 (4) (2009) 1052–1062.
[7] E. Aramendi, U. Irusta, E. Pastor, A. Bodegas, F. Benito, ECG spectral and morphological parameters reviewed and updated to detect adult and paediatric life-threatening arrhythmia, Physiol. Meas. 31 (6) (2010) 749.
[8] R. Fang, S. Pouyanfar, Y. Yang, S.-C. Chen, S. Iyengar, Computational health informatics in the big data age: A survey, ACM Comput. Surv. 49 (1) (2016) 12.
[9] J. Sun, C.K. Reddy, Big data analytics for healthcare, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013 1525–1525.
[10] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, MIT press Cambridge, 2016 http://www.deeplearningbook.org.
[11] F. Celesti, A. Celesti, L. Carnevale, A. Galletta, S. Campo, A. Romano, P. Bramanti, M. Villari, Big data analytics in genomics: The point on Deep Learning solutions, in: IEEE Symposium on Computers and Communications, ISCC, IEEE, 2017, pp. 306–309.
[12] O.E. Isafiade, O.E. Isafiade, A.B. Bagula, Data Mining Trends and Applications in Criminal Science and Investigations, first ed., IGI Global, Hershey, PA, USA, 2016, pp. 1–386.
[13] R.J. Martis, C. Chakraborty, A.K. Ray, A two-stage mechanism for registration and classification of ECG using Gaussian mixture model, Pattern Recognit. 42 (11) (2009) 2979–2988.
[14] M. Arif, et al., Robust electrocardiogram (ECG) beat classification using discrete wavelet transform, Physiol. Meas. 29 (5) (2008) 555.
[15] P. De Chazal, M. O'Dwyer, R.B. Reilly, Automatic classification of heartbeats using ECG morphology and heartbeat interval features, IEEE Trans. Biomed. Eng. 51 (7) (2004) 1196–1206.
[16] M. Llamedo, J.P. Martínez, Heartbeat classification using feature selection driven by database generalization criteria, IEEE Trans. Biomed. Eng. 58 (3) (2011) 616–625.
[17] C. Ye, B.V. Kumar, M.T. Coimbra, Heartbeat classification using morphological and dynamic features of ECG signals, IEEE Trans. Biomed. Eng. 59 (10) (2012) 2930–2941.
[18] G. Garcia, G. Moreira, D. Menotti, E. Luz, Inter-patient ECG heartbeat classification with temporal VCG optimized by PSO, Sci. Rep. 7 (1) (2017) 10543.
[19] G. De Lannoy, D. François, J. Delbeke, M. Verleysen, Weighted conditional random fields for supervised interpatient heartbeat classification, IEEE Trans. Biomed. Eng. 59 (1) (2012) 241–247.
[20] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, Physiobank, physiotoolkit, and physionet, Circulation 101 (23) (2000) e215–e220.
[21] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50.
[22] P. de Chazal, R.B. Reilly, A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features, IEEE Trans. Biomed. Eng. 53 (12) (2006) 2535–2543.

[23] T. Ince, S. Kiranyaz, M. Gabbouj, A generic and robust system for automated patient-specific classification of ECG signals, IEEE Trans. Biomed. Eng. 56 (5) (2009) 1415–1426.

[24] J. Wiens, J.V. Guttag, Active learning applied to patient-adaptive heartbeat classification, Adv. Neural Inf. Process. Syst. (2010) 2442–2450.

[25] Association for the Advancement of Medical Instrumentation and American National Standards Institute, Testing and Reporting Performance Results of Cardiac Rhythm and ST-segment Measurement Algorithms, ANSI/AAMI, The Association, 1999, pp. 1–36 https://books.google.it/books?id=gzPdtgAACAAJ.

[26] I. Christov, G. Gómez-Herrero, V. Krasteva, I. Jekova, A. Gotchev, K. Egiazarian, Comparative study of morphological and time-frequency ECG descriptors for heartbeat classification, Med. Eng. Phys. 28 (9) (2006) 876–887.

[27] Y.H. Hu, S. Palreddy, W.J. Tompkins, A patient-adaptable ECG beat classifier using a mixture of experts approach, IEEE Trans. Biomed. Eng. 44 (9) (1997) 891–900.

[28] D. Bianchi, P.D. Michele, C. Marchetti, B. Tirozzi, S. Cuomo, H. Marie, M. Migliore, Effects of increasing CREB-dependent transcription on the storage and recall processes in a hippocampal CA1 microcircuit, Hippocampus 24 (2) (2014) 165–177.

[29] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learning Res. 3 (Mar) (2003) 1157–1182.

[30] I. MathWorks, MATLAB and Statistics Toolbox Release 2014b, The MathWorks, Natick, MA, 2014.

[31] B. Chandrakar, O. Yadav, V. Chandra, A survey of noise removal techniques for ECG signals, Int. J. Adv. Res. Comput. Commun. Eng. 2 (3) (2013) 1354–1357.

[32] S. Cuomo, G. De Pietro, R. Farina, A. Galletti, G. Sannino, A revised scheme for real time ecg signal denoising based on recursive filtering, Biomed. Signal Process. Control 27 (2016) 134–144.

[33] G.M. Friesen, T.C. Jannett, M.A. Jadallah, S.L. Yates, S.R. Quint, H.T. Nagle, A comparison of the noise sensitivity of nine QRS detection algorithms, IEEE Trans. Biomed. Eng. 37 (1) (1990) 85–98.

[34] G. Garg, S. Gupta, V. Singh, J. Gupta, A. Mittal, Identification of optimal wavelet-based algorithm for removal of power line interferences in ECG signals, in: Power Electronics, IICPE, 2010 India International Conference on, IEEE, 2011, pp. 1–5.

[35] R. Farina, S. Dobricic, A. Storto, S. Masina, S. Cuomo, A revised scheme to compute horizontal covariances in an oceanographic 3D-VAR assimilation system, J. Comput. Phys. 284 (2015) 631–647.

[36] M.M. Al Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, R.R. Yager, Deep learning approach for active classification of electrocardiogram signals, Inform. Sci. 345 (2016) 340–354.

[37] W.K. Pratt, Digital Image Processing: PIKS Scientific Inside, Wiley, 2007 https://books.google.it/books?id=tpYHZO8111UC.

[38] R.B. Lake, Programs for digital signal processing, Proc. IEEE 69 (7) (1981) 856–857.

[39] I. Silva, G.B. Moody, An open-source toolbox for analysing and processing physionet databases in matlab and octave, J. Open Res. Softw. 2 (1) (2014) e27.

[40] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, IEEE Trans. Biomed. Eng. (3) (1985) 230–236.

[41] P. Laguna, New electrocardiographic signal processing techniques: Application to long-term records, (Ph.D. Thesis), Science Faculty, University of Zaragoza, 1990.

[42] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: large-scale machine learning on heterogeneous distributed systems 2016, pp. 1–19 arXiv preprint arXiv:1603.04467.

[43] A. Géron, Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems, OReilly Media, Sebastopol, 2017 https://books.google.it/books?id=bRpYDgAAQBAJ.

[44] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsletter 11 (1) (2009) 10–18.

[45] Y. Freund, R.E. Schapire, et al., Experiments with a new boosting algorithm, Icml 96 (1996) 148–156.

[46] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[47] G.H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[48] J. Rumelhart, Learning representations by back-propagation error, Nature 323 (9) (1986) 533–536.

[49] J.G. Cleary, L.E. Trigg, et al., K*: An instance-based learner using an entropic distance measure, in: Proceedings of the 12th International Conference on Machine Learning, Vol. 5, 1995, pp. 108–114.

[50] R.C. Holte, Very simple classification rules perform well on most commonly used datasets, Mach. Learn. 11 (1) (1993) 63–90.

[51] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, 1998, pp. 1–15.

[52] J.R. Quinlan, C4.5: Programs for Machine Learning, Elsevier, 2014 https://books.google.it/books?id=b3ujBQAAQBAJ.

[53] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[54] W.W. Cohen, Fast effective rule induction, in: Twelfth International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 115–123.

[55] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.

[56] S. Keerthi, S. Shevade, C. Bhattacharyya, K. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, Neural Comput. (3) (2001) 637–649.

[57] T. Hastie, R. Tibshirani, Classification by pairwise coupling, in: Advances in Neural Information Processing Systems, MIT Press, 1998, pp. 507–513.
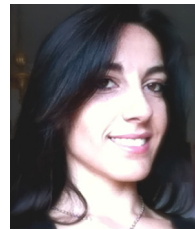
Giovanna Sannino received a master's degree in Telecommunications Engineering (Cum Laude) from the University of Naples Parthenope in April 2011 and a Ph.D. degree in Information Engineering at the University of Naples "Parthenope" in April 2015. In 2009, she earned a master degree, named "European Master on Critical Networked Systems", and a bachelor's degree in Computer Engineering in 2008 at the University of Naples Federico II. Currently, she is a researcher at Institute for High-Performance Computing and Networking (ICAR-CNR). Her research activities are focused in the area of mobile Health and Pervasive Computing. In addition, she is a IEEE 11073 Personal Health Device Working Group Member.

Giuseppe De Pietro is the director of ICAR-CNR. Current research interests are: pervasive and mobile computing, Clinical Decision Support Systems, Software architectures for e-health. He has been actively involved in many European and National projects, with industrial co-operations too. He is author of over 150 scientific papers published in international journals and conferences and he is involved in many program committees and journal editorial boards. He is Adjunct Professor in Temple University's College of Science and Technology, and IEEE and KES International Member.