Information Sciences 345 (2016) 340–354

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Deep learning approach for active classification of electrocardiogram signals

M.M. Al Rahhal[a], Yakoub Bazi[a,*], Haikel AlHichri[a], Naif Alajlan[a], Farid Melgani[b], R.R. Yager[c,1]

[a] *ALISR Laboratory, College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia*
[b] *Department of Information Engineering and Computer Science, University of Trento, Via Sommarive, 14, I-38123 Trento, Italy*
[c] *Machine Intelligence Institute, Iona College, New Rochelle, NY 10801, USA*

ARTICLE INFO

Article history:
Received 23 April 2015
Revised 1 December 2015
Accepted 29 January 2016
Available online 5 February 2016

Keywords:
ECG signal classification
Feature learning
Denoising autoencoder (DAE)
Deep neural network (DNN)
Active learning (AL)

ABSTRACT

In this paper, we propose a novel approach based on deep learning for active classification of electrocardiogram (ECG) signals. To this end, we learn a suitable feature representation from the raw ECG data in an unsupervised way using stacked denoising autoencoders (SDAEs) with sparsity constraint. After this feature learning phase, we add a softmax regression layer on the top of the resulting hidden representation layer yielding the so-called deep neural network (DNN). During the interaction phase, we allow the expert at each iteration to label the most relevant and uncertain ECG beats in the test record, which are then used for updating the DNN weights. As ranking criteria, the method relies on the DNN posterior probabilities to associate confidence measures such as entropy and Breaking-Ties (BT) to each test beat in the ECG record under analysis. In the experiments, we validate the method on the well-known MIT-BIH arrhythmia database as well as two other databases called INCART, and SVDB, respectively. Furthermore, we follow the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI) for class labeling and results presentation. The results obtained show that the newly proposed approach provides significant accuracy improvements with less expert interaction and faster online retraining compared to state-of-the-art methods.

1. Introduction

The Electrocardiogram (ECG) signal is a noninvasive test widely used for reflecting the underlying heart conditions. A careful inspection of its behavior is essential for detecting cardiac arrhythmias particularly in long-term recordings (usually over a period of 24 h). Therefore, the utilization of computer-based methods represents an important solution that can benefit cardiologists in the diagnosis.

In the last decades, several pattern recognition methods were developed for arrhythmia detection and classification [1–4,27,33,43]. Usually, these approaches are based on three main steps which are preprocessing, feature extraction; and classification. First, the ECG signals are enhanced by eliminating various kinds of noise and artifacts (i.e., baseline wanders,

[*] Corresponding author. Tel.: +966 506469808.
E-mail addresses: yakoub.bazi@gmail.com, ybazi@ksu.edu.sa (Y. Bazi), hhichri@ksu.edu.sa (H. AlHichri), najlan@ksu.edu.sa (N. Alajlan), melgani@disi.unitn.it (F. Melgani), yager@panix.com (R.R. Yager).
[1] Visiting Distinguished Scientist, King Saud University, Riyadh, Saudi Arabia.

http://dx.doi.org/10.1016/j.ins.2016.01.082

power line interference, and muscle contraction) [41,51,55,59,60]. After this step, the ECG waveforms, also known as PQRST, consisting mainly of P wave, QRS complex and T wave are extracted by means of segmentation [9,46,50]. Then several handcrafts features are calculated from these waveforms. In general, the available feature representation methods include, but are not limited to, morphology [14,18], temporal information [15,35], wavelet transform [61,63], high-order statistics (HOS) [35], Hermite basis function [37], and hidden Markov modeling (HMM) [12]. Then feature reduction techniques such as principal component analysis (PCA), independent component analysis (ICA), and linear discriminant analysis (LDA) are usually applied to reduce the dimensionality of the feature representation [42,62,63]. Finally, the obtained features are used to learn the decision function of a classifier such as neural networks (NN) [34], probabilistic NN [58], recurrent NN [17], support vector machines (SVMs) [1,27], least square SVM [19], path forest [40] and Gaussian processes (GPs) [1,44].

Despite these great efforts, it has been shown recently [14,27,64] that automatic methods do not perform well if the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI) for class labeling and results presentation are closely followed as a possible solution for standardization. Specifically, the AAMI standard defines five classes of interest: normal (N), ventricular (V), supraventricular (S), fusion of normal and ventricular (F) and unknown beats (Q). Regardless of the class definition, this standard essentially recommends for performance evaluation the adoption of inter-patient scenario (i.e., training and test ECG beats are extracted from different patients), which is not usually adopted in most of the works published in the literature. This requirement renders the automatic classification task very challenging due to the strong shift between the distributions of training and test subjects. Although, various handcrafted feature representations as well as many classifiers were considered as mentioned previously, the results obtained by automatic methods remain up till now unsatisfactory.

To overcome the above issues, semiautomatic methods allowing expert interaction are introduced as an alternative promising solution. Basically, these approaches start by training a global classifier on a large dataset and another local-classifier on the first few minutes from the test record labeled by an expert [5,15,41,32,34]. Then the outputs of both classifiers are fused using voting rules to classify the entire record. However, one major drawback of these methods lies in the selection scheme which does not take into consideration the importance of these beats in improving the classification accuracy. Indeed, it is not guaranteed that the selected first few minutes can efficiently model the statistical distribution of the data.

This paper proposes a novel approach for the active classification of ECG signals based on deep learning [8]. The idea of deep learning also known as feature learning (proposed for the first time by Hinton [25]) is about learning a good feature representation automatically from the input data [26,56,54,13,65]. Typical deep learning architectures include deep belief networks (DBNs) [26], stacked autoencoder (SAE) [54], convolutional neural networks (CNNs) [52]. Recently, compared to shallow architectures (i.e., handcrafted features fed as input to a kernel classifier), deep learning has shown outstanding results in many applications such as image classification [23], object recognition [6], face recognition [29], medical image analysis [10], and time series data [16,24,36]. For the particular case of physiological data, Mirowski et al. [45] used convolutional networks for epileptic seizure prediction from intracranial EEG signals. Längkvist et al. [41] proposed an RBM-based method for sleep stage classification from 4-channel polysomnography data. Wang and Shang [57] used DBN to automatically extract features from raw unlabeled physiological data. For the automatic classification ECG signals, one can find the solution proposed in [30] based on the combination DBN and SVM. In particular, DBN was used for feature learning and the obtained features are fed to SVM for training and classification.

In our context, we use deep learning to achieve two main objectives: (i) learn a suitable feature representation of the ECG signals in an automatic way unlike state-of-the-art methods which rely on handcrafted features; and (ii) use active learning (AL) techniques to reduce the expert effort in labeling data instances for inducing the classifier. Given the training data available at hand, we first learn an appropriate feature representation in an unsupervised way using a denoising autoencoder (DAE) with sparsity constraint. After this feature learning phase, we build an initial DNN tailored to the classification of AAMI classes by adding on the top of the resulting hidden representation layer a *softmax* regression layer. During the interaction phase, unlike the available methods, we do not make the user label the first few minutes but instead we use AL techniques.

The aim of AL is to rank the unlabeled set according to a criterion that allows us to select the most useful samples that can improve the model, thus minimizing the number of training samples necessary to maintain discrimination capabilities as high as possible. When faced with large amounts of unlabeled data, such algorithms automatically identify the exemplar beats for manual annotation [11,20–22,28,39,48,49]. The most ambiguous samples are given to the expert for labeling and then they are used to retrain the classifier. It is expected that the AL process will increase the generalization ability of the classification system on the difficult samples for the next iterations. As ranking criteria, the method relies on the DNN posterior probabilities to associate confidence measures such as entropy [21,28,48] and Breaking-Ties (BT) [11,39,49]. In the first criterion, we calculate for each ECG test beat the entropy value, and then the beats with the highest entropy values are selected for labeling. High values of entropy mean that the ECG beats are classified with low confidence, and thus adding them to the training set can be useful to improve the classifier decision regions in the feature space. In the second criterion called BT, the difference between the two highest DNN posterior probabilities is indicative of the way a sample is handled by the classifier. When the two highest values are close, the classifier confidence is low. Thus, the beats having low difference between the two highest support values are selected for labeling.

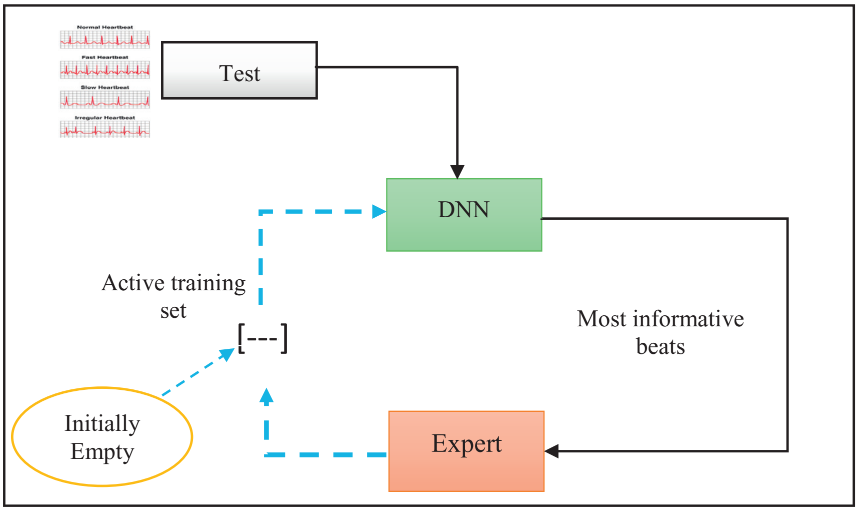

Fig. 1. Flowchart of the proposed method.

In the experiments, we validate the method on the well-known MIT-BIH arrhythmia database as well as two other databases called INCART and SVDB, respectively. The results obtained show that the newly proposed approach provides significant accuracy improvements with less expert interaction compared to state-of-the-art methods.

The rest of the paper is organized as follows. Detailed descriptions of the proposed approach are presented in Section 2. Experimental results are reported in Sections 3. Finally, conclusions and future directions are drawn in Section 4.

2. Description of the proposed approach

Let us consider $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ a training set composed of n ECG beats, where $\mathbf{x}_i \in \Re^D$ is the ECG beat vector of dimension D and $y_i \in [1,\ K]$ is its corresponding class label. Given this training set, we aim to classify an ECG record of a new patient unseen during the training phase and possibly obtained under different acquisition conditions using the learning paradigm shown in Fig. 1. In the next subsections, we provide a detail description of the proposed approach.

2.1. Phase 1: unsupervised feature learning using DAE

In this phase, we first identify a suitable deep architecture model to learn the underlying structure of the training data $\{\mathbf{x}_i\}_{i=1}^{n}$ available at hand. We use a denoising autoencoder (DAE) which is a symmetrical neural network mainly used for learning the features of a dataset in an unsupervised manner (see Fig. 2a) [16]. To build a robust feature representation, the DAE is trained to reconstruct the input $\mathbf{x}_i \in \Re^D$ from its noisy corrupted version $\tilde{\mathbf{x}}_i$**.** While DAE can be trained to perform dimensionally reduction like PCA; it is more useful to learn sparse representations by using a large number of hidden nodes. This means representing a high-dimensional original signal by using a few representative atoms on a low-dimensional manifold similar to sparse coding [53].

Typically the sparse DAE is made up of encoding and decoding parts, respectively. In the encoding part, the noisy input $\tilde{\mathbf{x}}_i$ is mapped to the hidden representation $\mathbf{h}_i \in \Re^L$ through the nonlinear activation function f as follows:

$$\mathbf{h}_i = f(\mathbf{W}^{(e)}.\widetilde{\mathbf{x}}_i + \mathbf{b}^{(e)} \tag{1}$$

where $\mathbf{W}^{(e)} \in \Re^{L\times D}$ is the encoder weight matrix and $\mathbf{b}^{(e)} \in \Re^L$ is the encoder bias vector. A typical choice of the activation function is the sigmoid function i.e, $f(v) = 1/(1+\exp(-v))$. In the decoding phase, the hidden representation is also mapped through a nonlinear activation function to reconstruct the input vector $\mathbf{x}_i$ as follows:

$$\mathbf{r}_i = f(\mathbf{W}^{(d)}.\mathbf{h}_i + \mathbf{b}^{(d)} \tag{2}$$

where $\mathbf{W}^{(d)} \in \Re^{D\times L}$ is the decoder weight matrix and $\mathbf{b}^{(d)} \in \Re^D$ is the decoder bias.

To determine the parameter vector $\boldsymbol{\theta}_{DAE} = \{\mathbf{W}^{(e)}, \mathbf{W}^{(d)}, \mathbf{b}^{(e)}, \mathbf{b}^{(d)}\}$ representing the complete DAE structure, we minimize the following cost function:

$$\mathcal{L}_1\left(\boldsymbol{\theta}_{DAE}\right) = \frac{1}{2n}\sum_{j=1}^{n} ||\mathbf{x}_i - \mathbf{r}_i||^2 + \frac{\lambda_1}{2}\left(||\mathbf{W}^{(e)}||_F^2 + ||\mathbf{W}^{(d)}||_F^2\right) + \lambda_2 \sum_{j=1}^{L} \mathrm{KL}\left(\rho||\hat{\rho}_j\right) \tag{3}$$

with

$$\mathrm{KL}\left(\rho\hat{\rho}_j\right) = \rho log\frac{\rho}{\hat{\rho}_j} + (1-\rho)log\frac{1-\rho}{1-\hat{\rho}_j} \tag{4}$$

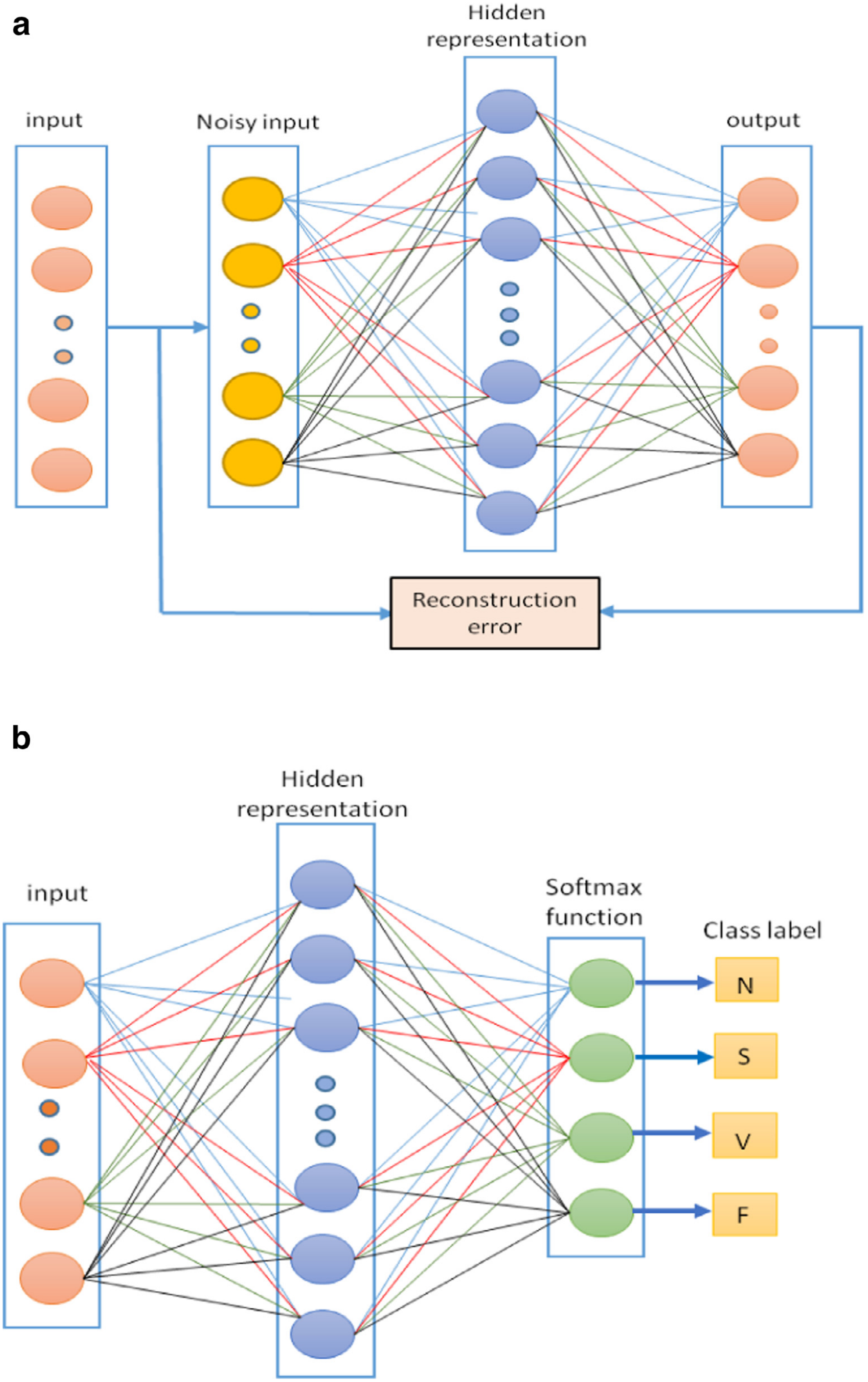

Fig. 2. DNN architecture: (a) pretraining using DAE, and (b) supervised fine tuining.

Where λ_1 and λ_2 are regularization parameters and $||\cdot||_F$ is the Frobenius norm. The first term in (3) represents the reconstruction error over all n training examples. The weight decay penalty term and the sparsity constraint are added in the cost function to avoid over-fitting; improve generalization ability, and yield sparse representations. The sparsity constraint is introduced through the Kullback–Leibler divergence KL$(\cdot)$ computed between two distributions with means ρ and $\hat{\rho}_j$. Here, ρ refers to the target activation function of the hidden nodes, and $\hat{\rho}_j = \frac{1}{n}\sum_{i=1}^{n}[\mathbf{h}_i]_j$ is the average activation function of the jth hidden node over all training examples. Typically, a non-redundant overcomplete representation will be learned as in sparse coding when ρ is set to a small value.

To optimize the cost function in (3), we first initialize the parameter vector $\boldsymbol{\theta}_{DAE}$ to small values near zero then we use the second order optimization method called L-BFGS [38] which is a quasi-Newton method based on the BFGS [47] update procedure. To reduce the number of parameters, the weights learned for the coding layer are simply tied to the decoding layer i.e. $\mathbf{W}^{(d)}$=$\mathbf{W}^{(e)^T}$ during the optimization process.

It is worth noting that in order to build a deep learning architecture with H hidden layers, H-DAEs are trained in a greedy layer-wise unsupervised mode. The main idea is to learn a hierarchy of features one level at a time. Specifically, the learning process starts by training the first DAE in unsupervised way by optimizing (3) with the original input data to obtain the first hidden representation layer. Then the reconstruction layer of this DAE is removed and the obtained hidden layer is used as the input data for training the next DAE to generate higher-level representations, and so on. Finally, the learned feature representations can be fed as input to a linear classifier such as SVM. However, this solution requires retraining the SVM classifier on a large training set (augmented at each AL iteration with samples labeled by the expert) which is computationally expensive and not practical in our context. For such purpose, this work considers an alternative computationally efficient approach as shown in the next phase.

2.2. Phase 2: supervised fine tuning

Once the greedy layer wise pretraining is complete, one can add on the top of the resulting hidden representation layers a *logistic/softmax* regression layer to perform binary/multiclass classification. This yields a DNN tailored to a task-specific supervised learning (see Fig. 2 b). In our case, we append the softmax regression layer as we are dealing with a multiclass classification problem. Then we fine tune the entire DNN using backpropagation by minimizing the following cost function [8]:

$$\mathcal{L}_2\left(\boldsymbol{\theta}_{DNN}\right) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} 1(y_i = k) log\left(\frac{\exp\left(h_{\boldsymbol{\theta}_{DNN}}(\mathbf{x}_i)\right)}{\sum_{k=1}^{K}\exp\left(h_{\boldsymbol{\theta}_{DNN}}(\mathbf{x}_i)\right)}\right) + \frac{\lambda_3}{2n}\left(||\mathbf{W}_{softmax}||_F^2 + \sum_{l=1}^{H}||\mathbf{W}_l||_F^2\right) \quad (5)$$

where $1(\cdot)$ is an indicator function that takes 1 if the statement is true otherwise it takes 0. The first term refers to the cross entropy loss for the softmax layer, the second term is again the weight decay penalty, and $h_{\boldsymbol{\theta}_{DNN}}(\mathbf{x}_i)$ is the output of the DNN for an input $\mathbf{x}_i$.

The estimation of the vector of parameters $\boldsymbol{\theta}_{DNN} = \{\mathbf{W}_1, \ldots, \mathbf{W}_H, \mathbf{b}_1, \ldots, \mathbf{b}_H, \mathbf{W}_{softmax}\}$ of the DNN starts by initializing the weights $\mathbf{W}_{softmax}$ of the softmax layer to small random values whereas the weights of the H hidden layers are initialized by the encoding weights obtained in the pretraining phase. Then the cost (5) is minimized with a min-batch gradient descent algorithm [52].

2.3. Phase 3: DNN fine tuning with AL

As mentioned in the introduction section, most of the works related to AAMI heart beat classification allow the expert to label the first few minutes of the record without taking into consideration the importance of these signals in boosting the classification accuracy. Here, we use an alternative solution based on AL which asks the expert to label the most relevant ECG test beats through an iterative process instead of labeling the first ECG beats. Basically, the idea of AL is to define appropriate criteria for ranking the ECG beats according to their relevance to the classification task. Then these beats are added to an active set (initially empty) for fine tuning the DNN classifier.

In order to select the most relevant beats, this work explores two different selection criteria based on entropy [21,28,48] and BT [11,39,49], respectively. In the first criterion, we calculate for each test beat the entropy score. Then the m_{AL} beats with the highest entropy values are selected for labeling. High values of entropy mean that the ECG beats are classified with low confidence, and thus adding them to the training set can be useful to improve the classifier decision regions in the feature space.

The second criterion called BT is based on the posterior probabilities of associating a sample to a given class. In a multiclass setting, the difference between the two highest posterior probabilities is indicative of the way a sample is handled by the classifier. When the two highest values are close, the classifier confidence is low. Thus, the m_{AL} beats having the lowest difference between the two highest posterior values are selected for labeling.

It is worth noting here that in order to avoid overfitting problems due to the inclusion of new uncertain beats in the active training set, we fuse the DNN outputs of the current and the previous iterations using the max rule. We found experimentally that this solution inspired from the concept of elitism in evolutionary computation [5] leads to more stable

Table 1
ECG databases used in the experiments.

Database	# Records	# Leads	Sample rate (Hz)
MIT-BIH	48 records, Each record is 30 min in length	2 leads	360
INCART	75 records. Each record is 30 min.	12 leads	257
SVDB	78 records. Each record is 30 min in length	2 leads	128

behaviors as we are updating the network with few labeled beats. The following algorithm provides the main steps of the proposed approach called Active-DNN.

```
Algorithm: Active-DNN
Input:
    - Training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
    - Test record: $Rec = \{\mathbf{x}_j\}_{j=1}^{m}$
    - DNN parameters $(\lambda_1,\ \lambda_2, \lambda_3,\ L)$
    - AL iterations: $ITER$
    - Number of ECG beats to label at each iteration: $m_{AL}$
Output:
    - Classification result
Step 1: Compute the weights $\boldsymbol{\theta}_{DAE}$ of DAE by optimizing the cost function (3)
        using L-BFGS (Phase 1);
Step 2: Use $\boldsymbol{\theta}_{DAE}$ to initialize DNN and compute $\boldsymbol{\theta}_{DNN}$ by optimizing the cost function (5) using backpropagation (Phase 2);
Step 3: Classify the test record Rec with the trained DNN; and let Post be the estimated posterior probabilities;
Step 4: Set the initial active training set to an empty set: $Tr_{AL} = \phi$;
Step 5: $for\ Iter = 1 : ITER$ (Phase 3)
    Step 5.1: Compute the uncertainty (i.e., BT or entropy) associated with each test beat using the estimated Post;
    Step 5.2: Rank the signals of the test record Rec based on their uncertainty;
    Step 5.3: Ask an expert to label the top ranked $m_{AL}$ signals;
    Step 5.4: Augment the active training set with these new labeled signals:
            $Tr_{AL} = [Tr_{AL}; Selected_{m_{AL}}]$;
    Step 5.5: Updated the weights of DNN by fine tuning on $Tr_{AL}$;
    Step 5.6: Classify the test record Rec with the updated DNN; and Let $Postnew$ be the current estimated posterior probabilities;
    Step 5.7. Use max rule to update the DNN posterior probabilities:
          $PostUpdated = \max(Post, Postnew)$;
    Step 5.8. Set $Post = PostUpdated$;
    Step 5.9. Set the labels of the test record Rec based on the maximum of $Post$;
end
Step 6: Final classification result.
```

3. Experimental results

3.1. Dataset description

In the experiments, we use three different ECG databases to evaluate the propose method as shown in Table 1. We recall that class Q (unclassified) is discarded since it is marginally represented in these three databases.

3.1.1. MIT-BIH arrhythmia Database (MIT-BIH)

This database consists of 48 two-lead recordings of approximately half-hour long for each record and sampled at 360 Hz. This database contains annotation for both beat class information and timing information verified by independent expert. The first 20 records (100–124) include representative beats to be included in the common training data. The remaining 24 used records (200–234) contain junctional, ventricular and supraventricular arrhythmias. Similar to [14], the four recordings with paced beats are discarded.

3.1.2. St.-Petersburg Institute of Cardiological Technics 12-lead arrhythmia database (INCART)

This database consists of 75 recordings which are annotated and extracted from 32 Holter records. Each record contains 12 standard leads and was collected from several patients (17 men and 15 women, aged between 18 and 80) undergoing tests for coronary artery disease. The duration of each record is 30 min and sampled at 257 Hz. The annotations were produced by an automatic algorithm, and then, corrected manually.

3.1.3. MITBIH supraventricular arrhythmia database (SVDB)

This database consists of 78 two-lead recordings of approximately 30 min and sampled at 128 Hz. The beat type annotations of the recordings were first automatically performed, by the Marquette Electronics 8000 Holter scanner and later reviewed and corrected by a medical student.

Table 2
Number of training and testing ECG beats used in the experiments.

	Database	N	S	V	F	#REC
Training	MIT-BIH (DS1)	45777	973	3769	414	22
Test	MIT-BIH (DS2)	44011	2049	3216	388	22
	INCART	153545	1958	2000	219	75
	SVDB	145436	10733	8281	23	70

3.2. Experiments setup and performance evaluation

Similar to [14,15], all ECG signals are first preprocessed using a 200 ms width median filter to remove P wave and QRS complex, then a 600 ms width median filter to remove T wave. The resulted signals are subtracted from the original signals to yield the baseline-corrected ECG signals. Then a 12-order low-pass filter with a 35 Hz cut-off frequency is applied to remove power-line and high-frequency noise. Then we build the initial training set from all ECG beats of the following 22 records of the MIT-BIH database grouped in DS1 = {101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230} [12]. The remaining records of this database, grouped in DS2 = {100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234} as well as all records of the other two databases are left for test as shown in Table 2.

To learn a suitable feature representation of the data with DAE, we use the complete ECG waveform (i.e., ECG morphology) of lead I. In addition, since it is well known that the temporal information is important for discerning the class signatures, we add four other temporal features [14,15]. These temporal features are: (1) the Pre-RR feature which is the distance between a current R peak and its previous R peak; (2) the Post-RR feature representing the distance between the current R peak and the next R peak; (3) the local RR interval computed by averaging all the RR intervals within a sliding window covering the past 10 seconds episode of the given ECG signal; and (4) the global average RR interval which is the average of RR intervals within a sliding window covering the past 5 min episode of the given ECG signal, which reflects the background rhythm information in the past 5 min episode.

To extract the ECG waveform as well as the above four temporal features, we perform the QRS detection and ECG wave boundary recognition tasks by means of the *ecgpuwave* software available on: http://www.physionet.org/physiotools/ecgpuwave/src/. Then we resample all segmented ECG signals to the same periodic length equal to 50 uniformly distributed samples. The length of the initial feature vector fed as input to DAE including morphology and temporal features equals 54 for each beat.

For performance evaluation, we present the results in terms of VEB (V class versus [N, S, and F]) and SVEB (S class versus [N, V, and F]). In particular, we use the standard measures: sensitivity(Se), positive predictive value (Pp), specificity (Sp), overall accuracy(OA) [14,34,35].

3.3. Generation of an initial DNN model (pretraining and fine tuning using DS1 of MIT-BIH)

First of all, we shall identify a suitable initial architecture of our DNN build on the basis of DAE and the softmax regression layer. We particularly, follow the practical recommendation of [7] of training deep architectures. To this end, we pre-train DAE on DS1 of MIT-BIH using the LB-FGS method then we use its results for initializing the DNN and then fine tune the whole architecture with backpropagation. Specifically for pre-training DAE, we initialize θ_{DAE} to small values in the range $[-0.005\ 0.005]$ randomly; we fix the target activation function of the hidden unit to $\rho = 0.05$ [24]; and then use the default parameters of the function *minFunc* [66] for optimizing the cost function (3). For the backpropagation algorithm, we use a mini-batch gradient optimization method (i.e., *learning rate* is set to 1, *momentum* to 0.5, and *batchsize* to 100). Then we search for the remaining DNN hyperparameters $(\lambda_1,\ \lambda_2, \lambda_3)$ using a 5-fold cross-validation technique in the range [0, 1]. We repeat this searching process, for different sizes of the hidden representation layer: $L = \{10,\ 25,\ 50,\ 75,\ 100,\ 200,\ 300\}$. We note that the experiments are carried out on a desktop with the following characteristics (core i7, CPU 2.50 GHz, RAM 16 GB, and GPU Nvidia Geoforce GTX 850 M).

Fig. 3 shows the training results obtained for each configuration on DS1. Here we clearly notice that using an overcomplete representation (similar to sparse coding) leads to better classification results compared to dimensionality reduction. Recall that the dimension of the original feature is equal to 54, thus the scenario with 100 hidden nodes is a reasonable choice for our initial DNN model. In Fig. 4, we also show the learned signatures for each AAMI class, and it can be observed that they are sparse and discriminative. It is important to note that with this architecture much of the computation burden is done in this phase only, while fine tuning the DNN weights during the AL process is computationally very efficient.

In order to show the superiority of DNN compared to standard NN, we repeat the above training process but without the pre-training phase. That is we train the complete architecture using only backpropagation. The results shown in Fig. 3 confirm clearly the superiority of DNN over standard NN.

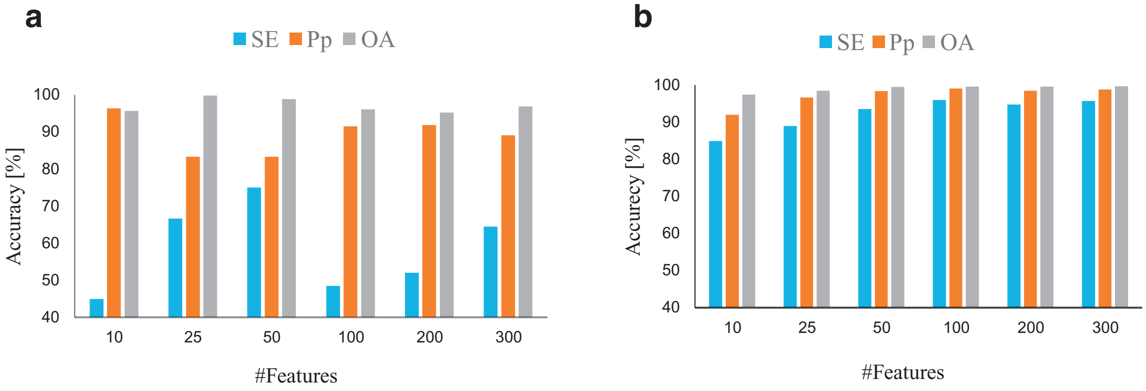

Fig. 3. Classification accuracies obtained on DS1 using: (a) standard NN; and (b) DNN.

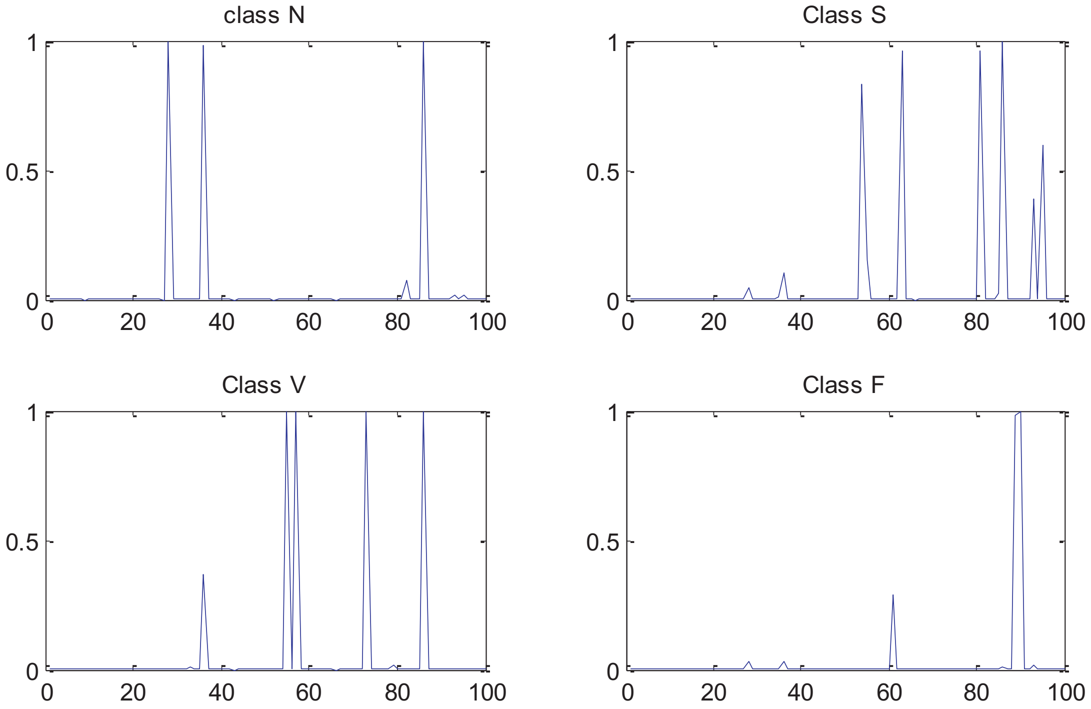

Fig. 4. Example of features learned by DNN for each AAMI class.

3.4. Results on MIT-BIH database

For this database, we present the results by considering three different scenarios for building the test set as usually done in the works dealing with the AAMI norm: (1) using the 11 common testing records for VEB {i.e. 200 202 210 213 214 219 221 228 231 233 234} and the 14 testing records for SVEB {i.e. 200 202 210 212 213 214 219 221 222 228 231 232 233 234}; (2) using the 24 common testing records from 200 up to 234; and (3) using All 48 records (i.e., DS1+DS2). To highlight the benefit of the using BT and entropy selection criteria, we also include for comparison purposes the results obtained by random selection and by labeling the first 300 samples from each record as done in the literature. In the rest of the paper, we call the corresponding deep learning schemes as DNN-BT, DNN-Entropy, DNN-RS, and DNN-First300.

Figs. 5 and 6 show an example of the behavior of $(OA, Se, \; and \; Pp)$ for VEB and SVEB, respectively versus the number of queries for DS1+DS2. We note that for each query the expert labels 10 ECG beats from the test record and then these labeled beats are added to the active training set (initially set to empty) in order to update the weights of DNN through backpropagation. As can be seen DNN-BT and DNN-Entropy clearly provide better results compared to DNN-RS and DNN-First300. For the first scenario, the initial values of $(OA, Se, \; Sp \; and \; Pp)$ for SEVB are equal to (91.3%, 11.5 %, 95.8%, and 13.2%). After adding 5 iterations (50 beats) per record, DNN-BT yields accuracies above 90% that is (93.1, 92.7%, 99.6% and

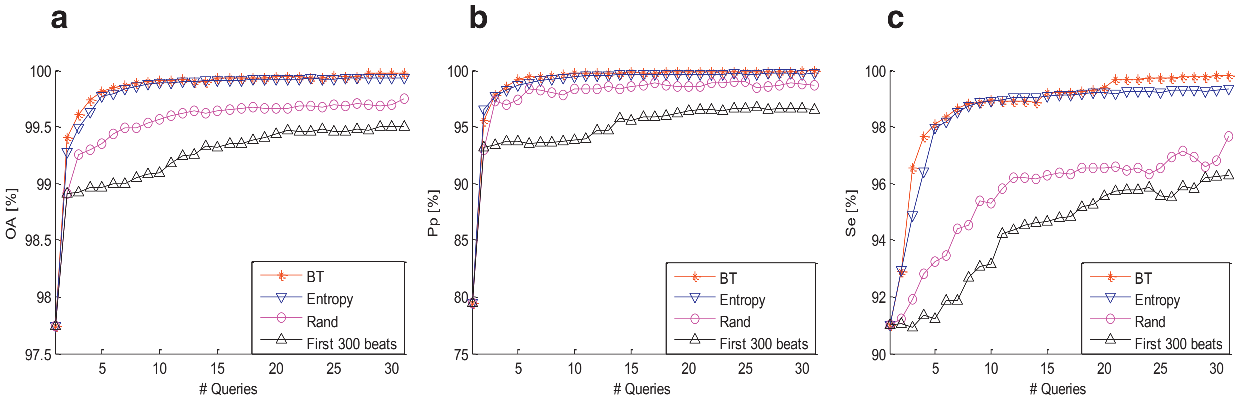

Fig. 5. VEB Classification results obtained on DS1 + DS2 of MIT-BIH: (a) *OA*; (b) *Pp*, and (c) *Se*.

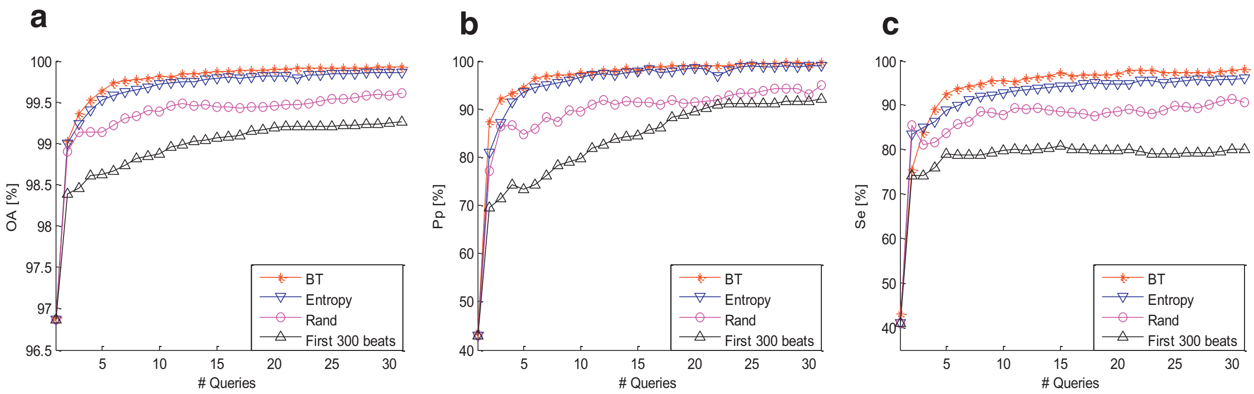

Fig. 6. SVEB Classification results obtained on DS1 + DS2 of MIT-BIH: (a) *OA*; (b) *Pp*, and (c) *Se*.

Table 3
Classification results in terms of VEB (11 common records) and SVEB (14 common records) of MIT-BIH.

Method	Labeled	SVEB				VEB			
		Se	*Pp*	*Sp*	*OA*	*Se*	*Pp*	*Sp*	*OA*
Chazal and Reilly [15]	500	87.7	47	N/A	95.9	94.3	96.2	N/A	99.4
Hu et al. [31]	300	N/A	N/A	N/A	N/A	78.9	75.8	96.8	94.8
Ince et al. [32]	300	81.8	63.4	98.5	96.1	90.3	92.2	98.8	97.9
Jaing et al. [34]	300	74.9	78.8	98.8	97.5	94.3	95.8	99.4	98.8
Deep-entropy	0	11.5	13.2	95.8	91.3	84.7	87.1	98.3	96.7
	50	86.3	91.7	99.6	98.9	96.3	98.3	99.6	99.4
	100	91.9	94.5	99.7	99.3	98.0	99.0	99.7	99.7
	200	93.8	98.4	99.9	99.6	97.8	99.5	99.8	99.7
	300	95.1	99.1	100.0	99.7	97.5	99.5	99.8	99.7
Deep-BT	0	11.5	13.2	95.8	91.3	84.7	87.1	98.3	96.7
	50	90.5	96.2	99.8	99.3	97.4	99.0	99.8	99.6
	100	93.8	98.8	99.9	99.6	98.5	99.4	99.9	99.8
	200	95.6	99.8	100.0	99.8	98.5	99.8	100.0	99.8
	300	98.1	99.7	100.0	99.9	99.4	99.8	100.0	99.8

99.3%), while, DNN-Entropy takes 10 iterations to provide similar results. On the other side, DNN-First300 exhibits the worst behavior as it provides an $(OA, Se,\ Sp\ and\ Pp)$ equal to (74.3%, 89.3%, 99.5% and 98.2%) after 30 iterations. For VEB, the initial values $(OA, Se,\ Sp\ and\ Pp)$ are equal to (96.7%, 84.7%, 98.3 and 87.1%). After 5 iterations, DNN-BT and DNN-Entropy reach accuracies of (98.6%, 92.4%, 99.29%, and 95.5%) and (98.9%, 91.2%, 99.8%, and 99.2%), respectively. Here again, DNN-First300 exhibits a less competing results compared to the other three learning modes. In Table 3, we report the results obtained by Deep-Entropy and Deep-BT, respectively. It is worth noting that the same observations can be easily recognized from Tables 4 and 5 related to the other scenarios. We recall that these results are obtained by applying the elitism trick to

Table 4
Classification results in terms of VEB and SVEB using 24 testing records of MIT-BIH.

Method	Labeled	SVEB				VEB			
		Se	*Pp*	*Sp*	*OA*	*Se*	*Pp*	*Sp*	*OA*
Jaing et al. [34]	300	50.6	67.9	98.8	96.6	86.6	93.3	99.3	98.1
Ince et al. [32]	300	62.1	56.7	98.5	96.1	83.4	87.4	98.1	97.6
Deep-entropy	0	37.8	40.5	97.5	94.9	90.1	87.1	98.6	97.8
	50	87.3	88.0	99.5	98.9	94.5	96.2	99.5	99.1
	100	91.5	90.7	99.6	99.2	95.7	98.4	99.7	99.4
	200	92.1	93.5	99.7	99.4	97.2	98.9	99.8	99.6
	300	92.4	93.1	99.7	99.4	97.2	98.8	99.8	99.6
Deep-BT	0	37.8	40.5	97.5	94.9	90.1	87.1	98.6	97.8
	50	91.7	93.9	99.7	99.4	97.3	98.8	99.8	99.6
	100	94.4	97.5	99.9	99.7	98.5	99.2	99.8	99.8
	200	96.1	98.4	99.9	99.8	98.8	99.6	99.9	99.8
	300	96.7	99.3	100.0	99.8	98.8	99.7	99.9	99.9

Table 5
Classification results in terms of VEB and SVEB using 44 testing records (DS1 + DS2) of MIT-BIH.

Method	Labeled	SVEB				VEB			
		Se	*Pp*	*Sp*	*OA*	*Se*	*Pp*	*Sp*	*OA*
Ince et al. [32]	300	63.2	53.7	99	97.4	84.6	87.4	98.7	98.3
	0	41.0	42.9	98.5	96.9	91.0	79.5	98.2	97.7
Deep-entropy	50	90.0	94.5	99.9	99.6	98.2	98.8	99.9	99.8
	100	93.2	97.0	99.9	99.7	98.9	99.5	99.9	99.9
	200	94.8	98.3	100.0	99.8	99.2	99.6	99.9	99.9
	300	95.9	98.9	100.0	99.9	99.3	99.7	99.9	99.9
Deep-BT	0	41.0	42.9	98.5	96.9	91.0	79.5	98.2	97.7
	50	93.7	96.3	99.9	99.7	98.3	99.5	99.9	99.9
	100	95.3	97.6	99.9	99.8	98.9	99.7	100.0	99.9
	200	97.8	99.0	100.0	99.9	99.7	99.9	100.0	99.9
	300	99.0	99.6	100.0	99.9	99.8	99.9	99.9	100.0

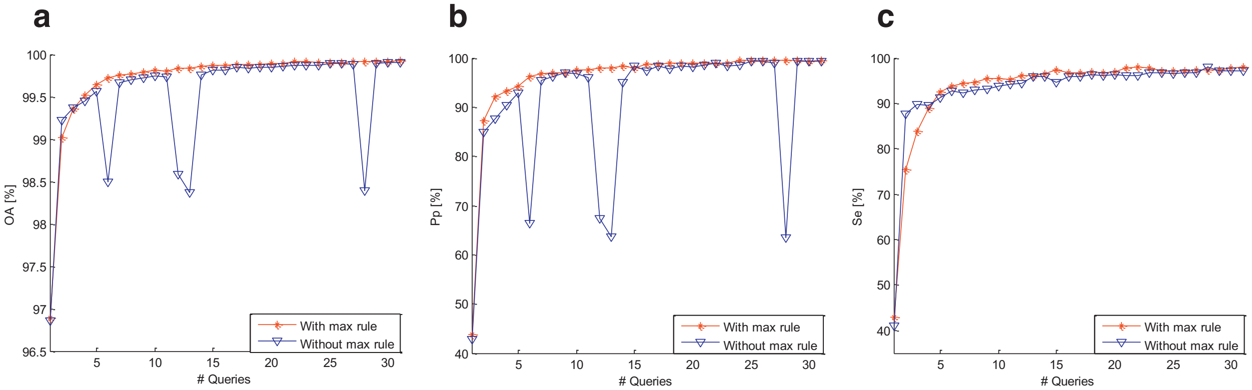

Fig. 7. Effect of the max rule on the stability of DNN during the interaction phase: (SVEB results for DS1 + DS2 obtained without/with max rule): (a) *OA*; (b) *Pp*, and (c) *Se*.

the previous and current posterior probabilities generated by DNN. As seen clearly in Fig. 7 this trick allows filtering out unstable behaviors caused by adding the most ambiguous beats to the active training set.

From the above results one can draw the following conclusions: (i) the selection of ECG beats for labeling has direct impact on the classification accuracy; (ii) the BT and entropy selection criteria allow obtaining stable and better results compared to simple selection schemes based on random selection or labeling the first 300 beats as is usually done in the literature; (iii) The results shown in Tables 3–5 for all three scenarios, respectively clearly confirm the superiority of the proposed approach compared to state-of-the-art methods.

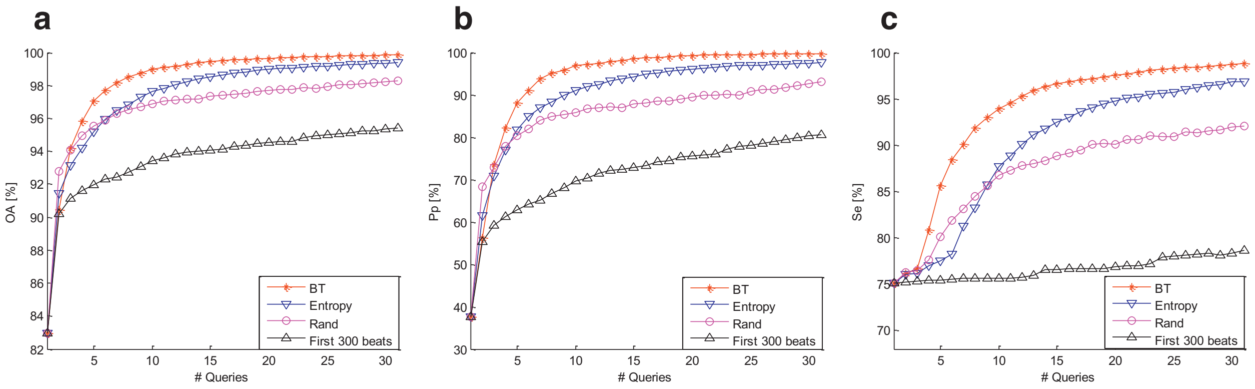

Fig. 8. VEB Classification results obtained on the INCART database: (a) *OA*; (b) *Pp*, and (c) *Se*.

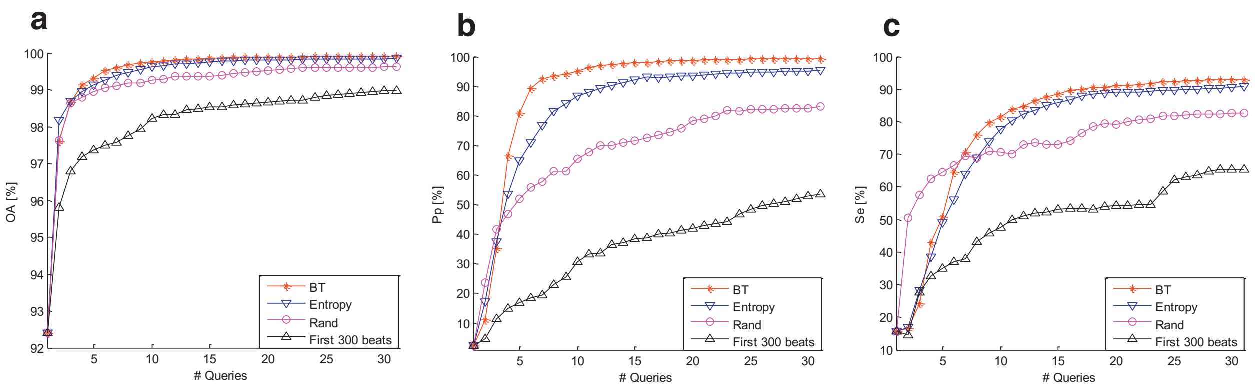

Fig. 9. SVEB Classification results obtained on the INCART database: (a) *OA*; (b) *Pp*, and (c) *Se*.

Table 6
Classification results obtained on the INCART database.

Method	Labeled	SVEB				VEB			
		Se	*Pp*	*Sp*	*OA*	*Se*	*Pp*	*Sp*	*OA*
Deep-entropy	0	15.58	2.54	93.26	92.40	75.11	37.63	83.98	82.99
	50	56.08	70.84	99.74	99.25	78.20	84.99	98.21	95.95
	100	80.13	88.05	99.88	99.66	88.86	91.88	98.97	97.84
	200	89.12	93.62	99.93	99.81	95.03	96.34	99.52	99.02
	300	90.86	95.34	99.95	99.85	96.88	97.51	99.67	99.36
Deep-BT	0	15.58	2.54	93.26	92.40	75.11	37.63	83.98	82.99
	50	64.61	89.15	99.91	99.52	88.43	91.04	98.86	97.69
	100	83.91	96.25	99.96	99.78	94.59	97.18	99.62	99.07
	200	91.22	98.78	99.99	99.89	97.66	99.37	99.91	99.66
	300	92.85	99.34	99.99	99.91	98.78	99.77	99.96	99.83

3.5. Results on INCART database

To assess further the generalization ability of the proposed approach, we repeat the above experiments on the INCART database. Recall that the results reported here are obtained using the initial DNN model trained on DS1 of MIT-BIH. Figs. 8 and 9 show the behavior of $(OA, Se, \ Sp \ and \ Pp)$ versus the number of queries for this dataset. Table 6 shows that the initial values of $(OA, Se, \ Sp \ and \ Pp)$ were low before starting the AL process as they are equal to (92.40%, 15.85%, 93.26%, and 2.54%) for SVEB and (82.99%, 75.11%, 83.99%, and 37.63%) for VEB. After adding 300 samples per record (30 iterations), DNN-BT reaches accuracies of (99.86%, 92.59%, 99.95% and 92.80%) for SVEB and (99.40%, 97.23%, 99.67, and 97.49%) for VEB, followed by DNN-Entropy which yields (99.85%, 90.86%, 99.95%, and 95.34%) for SVEB and (99.36%, 96.88%, 99.67%, and 97.51%) for VEB. Here again, we notice that DNN-First300 exhibits the worst behavior.

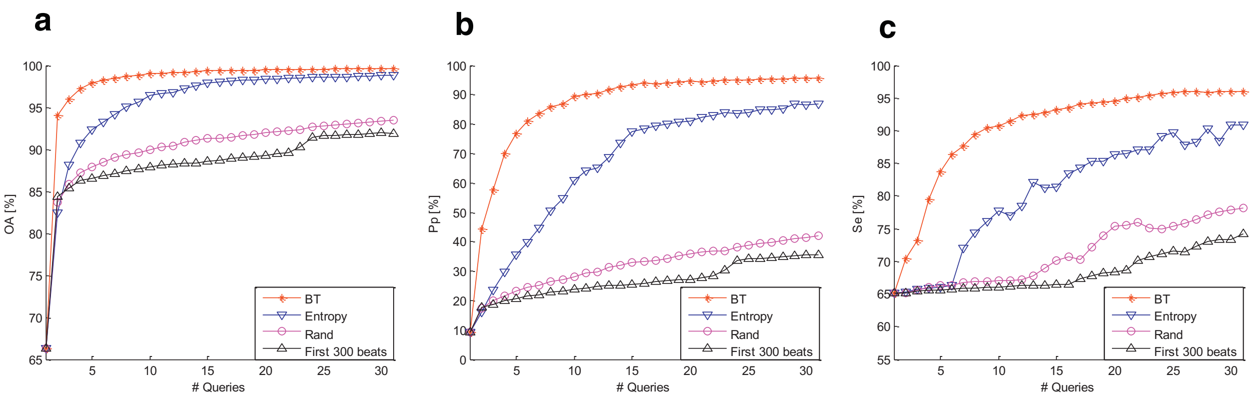

Fig. 10. VEB Classification results obtained on the SVDB database: (a) *OA*; (b) *Pp*, and (c) *Se*.

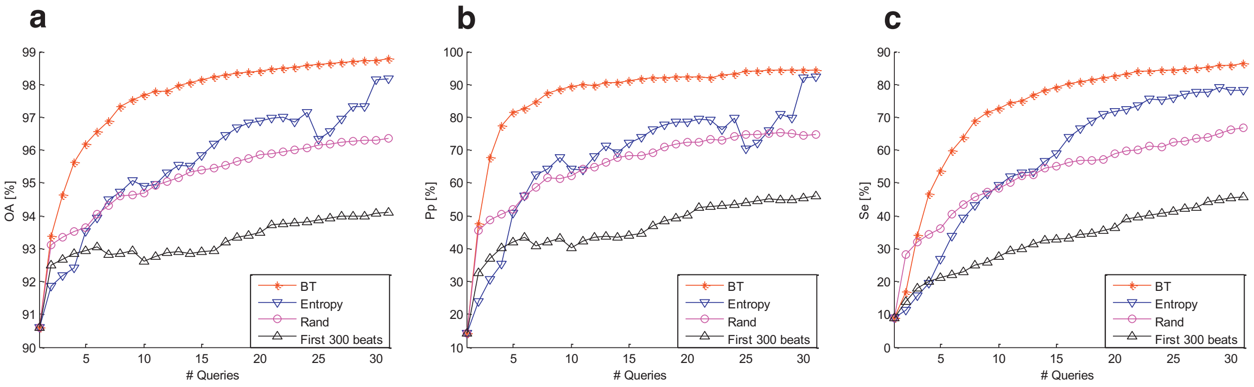

Fig. 11. SVEB Classification results obtained on the SVDB database: (a) *OA*; (b) *Pp*, and (c) *Se*.

Table 7
Classification results obtained on the SVDB database.

Method	Labeled	SVEB				VEB			
		Se	***Pp***	***Sp***	***OA***	***Se***	***Pp***	***Sp***	***OA***
Deep-entropy	0	8.80	14.31	96.32	90.61	65.19	9.31	66.32	66.27
	50	33.79	55.90	98.14	93.94	66.28	39.67	94.65	93.23
	100	51.83	63.87	97.95	94.94	76.92	64.08	97.71	96.67
	200	72.22	79.33	98.69	96.96	86.51	82.23	99.01	98.38
	300	78.28	92.38	99.55	98.16	90.81	86.85	99.27	98.84
Deep-BT	0	8.80	14.31	96.32	90.61	65.19	9.31	66.32	66.27
	50	59.78	82.58	99.12	96.55	86.27	81.03	98.93	98.29
	100	74.43	89.83	99.41	97.78	91.41	90.19	99.47	99.07
	200	83.15	92.28	99.51	98.45	95.01	94.48	99.71	99.47
	300	86.39	94.30	99.64	98.77	96.03	95.68	99.77	99.58

3.6. Results on SVDB database

Figs. 10 and 11 show the results of $(OA, Se,\ Sp\ and\ Pp)$ for VEB and SVEB, respectively. As can be seen, DNN-BT exhibits clearly a stable behavior compared to the other learning modes. Before starting the interaction process, the $(OA, Se,\ Sp,\ and\ Pp)$ are equal to (90.61%, 8.80%, 96.32%, and 14.31%) and (66.27%, 65.19%, 66.32%, and 9.31%) for SVEB and VEB respectively. After adding 300 samples per record, DNN-BT yields (98.11%, 79.04%, 99.44%, and 90.75%) for SVEB and (98.71%, 92.73, 99.02%, and 83.42%) for VEB. The worst result is again achieved by DNN-300. In Table 7, we summarize the main results obtained using these two learning schemes.

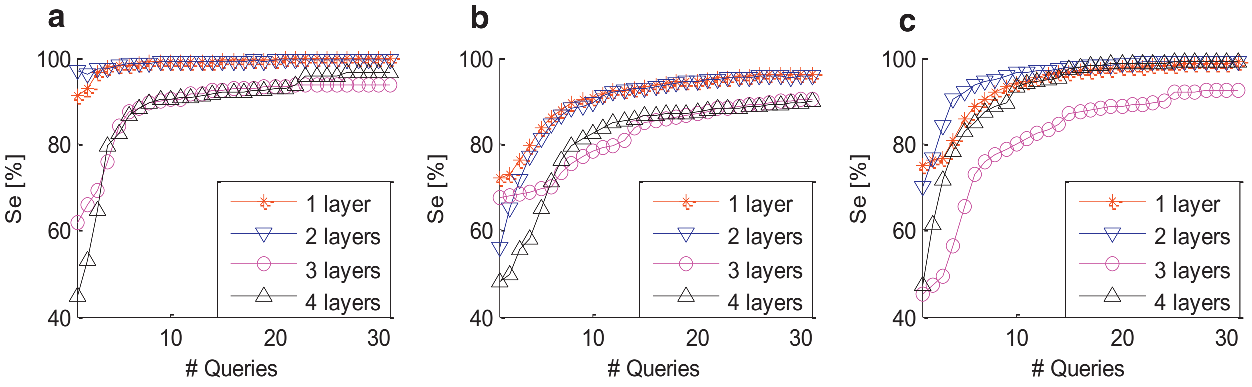

Fig. 12. $\mathrm{Se_{VEB}}$ classification results obtained by DNN-BT with multiple hidden layers on: (a) DS1 + DS2 of MIT-BIH; (b) SVDB; and (c) INCART databases.

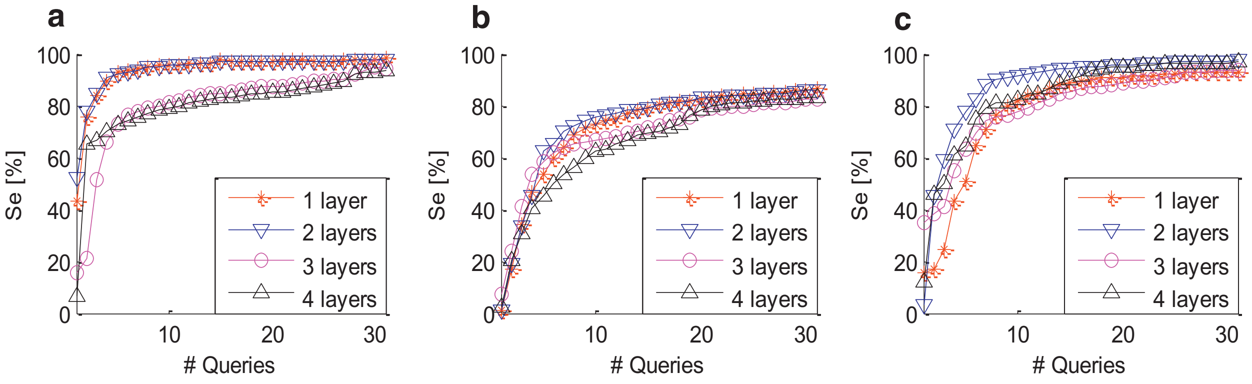

Fig. 13. $\mathrm{Se_{SVEB}}$ classification results obtained by DNN-BT with multiple hidden layers on: (a) DS1+DS2 of MIT-BIH; (b) SVDB; and (c) INCART databases.

3.7. *Results obtained by DNN-BT with multiple hidden layers*

To asses further the capability of the proposed method, we repeat the above experiments but with a configuration of multiple hidden layers. In particular, we consider scenarios characterized by $H=2$, 3, and 4 hidden layers. We use {100, 50} and {100, 50, 100} and {100, 50, 100, 50} nodes for each configuration, that is alternating between a sparse and dense representations. Here we recall that we tried several values for the hidden nodes but the best performances were obtained for the above values. Figs. 12 and 13 show the results obtained for the three databases. As can be seen, the configurations based on one and two hidden layers are both competing in terms of classification accuracy. By contrast, increasing the number of hidden layers leads to less accurate results. Such behavior suggests that updating the network with few training samples as done in our context (AL setting) seems to be not suitable for complex architectures.

4. Conclusions

In this paper, we have proposed a novel approach based on deep learning for the active classification of ECG signals. Compared to state-of-the art methods based on shallow architectures this approach has several desirable proprieties: (i) it learns automatically an appropriate sparse feature representation from the raw ECG using DAE; (ii) it relies on AL criteria for selecting the most valuable ECG beats for inducing the DNN classifier. The experimental results obtained on three different ECG databases showed that the proposed method is robust and computationally efficient during the iterative labeling process and clearly outperforms state-of-the-art methods. For future developments, we plan to increase the classification accuracy and reduce the expert interaction in many ways. For example: (i) the exploration of other deep architectures based on RBM and CNN; and (ii) the definition and the combination of AL criteria suitable for deep architectures.

Acknowledgments

The authors would like to acknowledge the support from the Distinguished Scientist Fellowship Program at King Saud University.

References

[1] N. Alajlan, Y. Bazi, F. Melgani, S. Malek, M.A. Bencherif, Detection of premature ventricular contraction arrhythmias in electrocardiogram signals with kernel methods, Signal Image Video Process. 8 (2014) 931–942.

[2] R. Alcaraz, F. Sandberg, L. Soŕnmo, J.J. Rieta, Classification of paroxysmal and persistent atrial fibrillation in ambulatory ECG recordings, IEEE Trans. Biomed. Eng. 58 (2011) 1441–1449.

[3] F. Alonso-Atienza, E. Morgado, L. Fernandez-Martinez, A. Garcia-Alberola, J.L. Rojo-Alvarez, Detection of life-threatening arrhythmias using feature selection and support vector machines, IEEE Trans. Biomed. Eng. 61 (2014) 832–840.

[4] A.S. Alvarado, C. Lakshminarayan, J.C. Principe, Time-based compression and classification of heartbeats, IEEE Trans. Biomed. Eng. 59 (2012) 1641–1648.

[5] C.W. Ahn, R.S. Ramakrishna, Elitism-based compact genetic algorithms, IEEE Trans. Evol. Comput. 7 (2003) 367–385.

[6] J. Bai, Y. Wu, J. Zhang, F. Chen, Subset based deep learning for RGB-D object recognition, Neurocomputing 165 (2015) 280–292.

[7] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: G. Montavon, G.B. Orr, K.-R. Müller (Eds.), Neural Network Tricks Trade, Springer Berlin Heidelberg, 2012, pp. 437–478.

[8] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[9] V. Bono, E.B. Mazomenos, T. Chen, J.A. Rosengarten, A. Acharyya, K. Maharatna, et al., Development of an automated updated Selvester QRS scoring system using SWT-based QRS fractionation detection and classification, IEEE J. Biomed. Health Inf. 18 (2014) 193–204.

[10] T. Brosch, R. Tam, Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2D and 3D images, Neural Comput. 27 (2015) 211–227.

[11] R.M. Castro, R.D. Nowak, Minimax bounds for active learning, IEEE Trans. Inf. Theory 54 (2008) 2339–2353.

[12] P.-C. Chang, J.-J. Lin, J.-C. Hsieh, J. Weng, Myocardial infarction classification with multi-lead ECG using hidden Markov models and Gaussian mixture models, Appl. Soft Comput. 12 (2012) 3165–3175.

[13] J.C. Li, Wing W.Y. Ng, Daniel S. Yeung, Patrick P.K. Chan, Bi-firing deep neural networks, Int. J. Mach. Learn. Cybern. 5 (1) (2014) 73–83.

[14] 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 (2004) 1196–1206.

[15] P. De Chazal, R.B. Reilly, A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features, IEEE Trans. Biomed. Eng. 53 (2006) 2535–2543.

[16] J. Deng, Z. Zhang, F. Eyben, B. Schuller, Autoencoder-based unsupervised domain adaptation for speech emotion recognition, IEEE Signal Process. Lett. 21 (2014) 1068–1072.

[17] E. Derya Übeyli, Recurrent neural networks employing Lyapunov exponents for analysis of ECG signals, Expert Syst. Appl. 37 (2010) 1192–1199.

[18] S.-M. Dima, C. Panagiotou, E.B. Mazomenos, J.A. Rosengarten, K. Maharatna, J.V. Gialelis, et al., On the detection of myocadial scar based on ECG/VCG analysis, IEEE Trans. Biomed. Eng. 60 (2013) 3399–3409.

[19] S. Dutta, A. Chatterjee, S. Munshi, Correlation technique and least square support vector machine combine for frequency domain based ECG beat classification, Med. Eng. Phys. 32 (2010) 1161–1169.

[20] Y. Fu, B. Li, X. Zhu, C. Zhang, Active learning without knowing individual instance labels: A pairwise label homogeneity query approach, IEEE Trans. Knowl. Data Eng. 26 (2014) 808–822.

[21] P.H. Gosselin, M. Cord, Active learning methods for interactive image retrieval, IEEE Trans. Image Process. 17 (2008) 1200–1211.

[22] M.M. Haque, L.B. Holder, M.K. Skinner, D.J. Cook, Generalized query-based active learning to identify differentially methylated regions in DNA, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (2013) 632–644.

[23] M. Hayat, M. Bennamoun, S. An, Deep reconstruction models for image set classification, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 713–727.

[24] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (2012) 82–97.

[25] G.E. Hinton, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.

[26] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.

[27] M.R. Homaeinezhad, S.A. Atyabi, E. Tavakkoli, H.N. Toosi, A. Ghaffari, R. Ebrahimpour, ECG arrhythmia recognition via a neuro-SVM–KNN hybrid classifier with virtual QRS image-based geometrical features, Expert Syst. Appl. 39 (2012) 2047–2058.

[28] S.-J. Huang, R. Jin, Z.-H. Zhou, Active learning by querying informative and representative examples, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 1936–1949.

[29] Z. Huang, R. Wang, S. Shan, X. Chen, Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning, Pattern Recognit. 48 (2015) 3113–3124.

[30] M. Huanhuan, Z. Yue, Classification of electrocardiogram signals with deep belief networks, in: Proceedings of the 2014 IEEE Seventeenth International Conference on Computer Science Engineering CSE, 2014, pp. 7–12.

[31] 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 (1997) 891–900.

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

[33] M. Javadi, S.A.A.A. Arani, A. Sajedin, R. Ebrahimpour, Classification of ECG arrhythmia by a modular neural network based on mixture of experts and negatively correlated learning, Biomed. Signal Process. Control 8 (2013) 289–296.

[34] W. Jiang, S.G. Kong, Block-based neural networks for personalized ECG signal classification, IEEE Trans. Neural Netw. 18 (2007) 1750–1761.

[35] Y. Kutlu, D. Kuntalp, Feature extraction for ECG heartbeats using higher order statistics of WPD coefficients, Comput. Methods Progr. Biomed. 105 (2012) 257–267.

[36] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24.

[37] G. de Lannoy, D. Francois, J. Delbeke, M. Verleysen, Weighted conditional random fields for supervised interpatient heartbeat classification, IEEE Trans. Biomed. Eng. 59 (2012) 241–247.

[38] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1989) 503–528.

[39] T. Luo, K. Kramer, D.B. Goldgof, Active learning to recognize multiple types of plankton, J. Mach. Learn. Res. 6 (2005) 589–613.

[40] E.J. da S. Luz, T.M. Nunes, V.H.C. de Albuquerque, J.P. Papa, D. Menotti, ECG arrhythmia classification based on optimum-path forest, Expert Syst. Appl. 40 (2013) 3561–3573.

[41] M. Langkvist, L. Karlsson, A. Loutfi, Sleep stage classification using unsupervised feature learning, Adv. Artif. Neural Syst. (2012) e107046.

[42] R.J. Martis, U.R. Acharya, L.C. Min, ECG beat classification using PCA, LDA, ICA and discrete wavelet transform, Biomed. Signal Process. Control 8 (2013) 437–448.

[43] F. Melgani, Y. Bazi, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, IEEE Trans. Inf. Technol. Biomed. 12 (2008) 667–677.

[44] F. Melgani, Y. Bazi, Detecting premature ventricular contractions in ECG signals with Gaussian processes, Comput. Cardiol. (2008) 237–240.

[45] P.W. Mirowski, Y. LeCun, D. Madhavan, R. Kuzniecky, Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG, in: Proceedings of the IEEE Workshop on Machine Learning Signal Process, 2008 MLSP 2008, 2008, pp. 244–249.

[46] X. Ning, I.W. Selesnick, ECG enhancement and QRS detection based on sparse derivatives, Biomed. Signal Process. Control 8 (2013) 713–723.

[47] J. Nocedal, Updating quasi-Newton matrices with limited storage, Math. Comput. 35 (1980) 773–782.

[48] E. Pasolli, F. Melgani, Active learning methods for electrocardiographic signal classification, IEEE Trans. Inf. Technol. Biomed. 14 (2010) 1405–1416.

[49] E. Pasolli, F. Melgani, Y. Bazi, Support vector machine active learning through significance space construction, IEEE Geosci. Remote Sens. Lett. 8 (2011) 431–435.
[50] P. Phukpattaranont, QRS detection algorithm based on the quadratic filter, Expert Syst. Appl. 42 (2015) 4867–4877.
[51] R. Sameni, M.B. Shamsollahi, C. Jutten, G.D. Clifford, A nonlinear Bayesian filtering framework for ECG Denoising, IEEE Trans. Biomed. Eng. 54 (2007) 2172–2185.
[52] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117.
[53] X. Sun, N.M. Nasrabadi, T.D. Tran, Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints, IEEE Trans. Geosci. Remote Sens. 53 (2015) 4457–4471.
[54] P. Swietojanski, A. Ghoshal, S. Renals, Convolutional neural networks for distant speech recognition, IEEE Signal Process. Lett. 21 (2014) 1120–1124.
[55] B.H. Tracey, E.L. Miller, Nonlocal means denoising of ECG signals, IEEE Trans. Biomed. Eng. 59 (2012) 2383–2386.
[56] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the Twenty Fifth International Conference on Machine Learning, New York, NY, USA, ACM, 2008, pp. 1096–1103.
[57] D. Wang, Y. Shang, Modeling physiological data with deep belief networks, Int. J. Inf. Educ. Technol. IJIET 3 (2013) 505–511.
[58] J.-S. Wang, W.-C. Chiang, Y.-L. Hsu, Y.-T.C. Yang, ECG arrhythmia classification using a probabilistic neural network with a feature reduction method, Neurocomputing 116 (2013) 38–45.
[59] J. Wang, Y. Ye, X. Pan, X. Gao, Parallel-type fractional zero-phase filtering for ECG signal denoising, Biomed. Signal Process. Control. 18 (2015) 36–41.
[60] S.K. Yadav, R. Sinha, P.K. Bora, Electrocardiogram signal denoising using non-local wavelet transform domain filtering, IET Signal Process. 9 (2015) 88–96.
[61] H. Yang, C. Kan, G. Liu, Y. Chen, Spatiotemporal differentiation of myocardial infarctions, IEEE Trans. Autom. Sci. Eng. 10 (2013) 938–947.
[62] C. Ye, B.V.K.V. Kumar, M.T. Coimbra, Heartbeat classification using morphological and dynamic features of ECG signals, IEEE Trans. Biomed. Eng. 59 (2012) 2930–2941.
[63] S.N. Yu, K.T. Chou, Integration of independent component analysis and neural networks for ECG beat classification, Expert Syst. Appl. 34 (2008) 2841–2846.
[64] Z. Zhang, J. Dong, X. Luo, K.-S. Choi, X. Wu, Heartbeat classification using disease-specific feature selection, Comput. Biol. Med. 46 (2014) 79–89.
[65] J. Zhang, S. Ding, N. Zhang, Z. Shi, Incremental extreme learning machine based on deep feature embedded, Int. J. Mach. Learn. Cybern. 7 (2016) 111–120.
[66] minFunc - unconstrained differentiable multivariate optimization in Matlab, 2005 (n.d.).