

Deep Learning-Based Classification of Massive Electrocardiography Data

Yan Yan, Xingbin Qin, Jianping Fan, and Lei Wang

Abstract—A deep learning model based approach for the classification of electrocardiography arrhythmia analysis was presented. Based on massive heartbeat samples extracted from ambulatory ECG database, a stacked autoencoder deep neural network model with pre-training and fine-tuning training method is adopted to classify the heartbeat. The labels were all based on the ANSI/AAMI EC57: 1998/(R)2008 standard: normal heartbeat, supraventricular ectopic heartbeat, ventricular ectopic heartbeat, fusion heartbeat. As well, the labeled MIT-BIH arrhythmia and MIT-BIH long term ECG databases were involved, which are regarded as the standard reference for the classifier. The stacked autoencoder algorithm was adopted for feature learning process instead of complex features extraction selection. From the massive unlabelled dataset, features in the ECG samples were acquired. A significant improvement of heartbeat classification performance is attained that the accuracy reached 99.34% and ventricular ectopic abnormality sensitivity accomplished 98.31%. With its self-learning ability, the proposed method has great potential to generalize to other related applications with a large amount of unlabelled data from the long-term clinical monitoring and healthcare monitoring.

Index Terms—electrocardiography classification, electrocardiography annotation, autoencoder, deep learning.

I. INTRODUCTION

AN era of big data in healthcare is now under way, decades of progress in digitising medical records accumulate vast amounts of medical data, simultaneously mobile healthcare and wearable sensor technologies offer healthcare data from larger population coverage. The noninvasive, inexpensive and well-established technology of electrocardiographic signal in mobile health or personal health has the greatest popularity in heart function analysis. Automated electrocardiography classification provides indispensable assist in long-term clinical monitoring, and a large number of approaches have been proposed for the task, easing the diagnosis of arrhythmic changes as well as further inspection, e.g., heart rate variability or heart turbulence analysis [1].

Y. Yan is with the Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China-mail: (yan.yan@siat.ac.cn).

X. Qin is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

J. Fan is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

L. Wang is with the Shenzhen Key Laboratory for Low-cost Healthcare, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. No. 1068, Xueyuan Road, Nanshan District, Shenzhen, Guangdong Province, China.

Manuscript received September 29, 2014; revised

Lots of algorithms had been proposed for the classification and detection of electrocardiography signals. The electrocardiography classification or detection task had been divided into two parts: the feature extraction process and classifier. Simple classifier such as linear discriminants [2] and kNN [3], more complex classifiers like neural networks [4]–[7], fuzzy inference engines [7], [8], hidden Markov model [9], [10], independent component analysis [11] and support vector machine [3], [12], [13] were also adopted by lots of researchers.

Beyond the classifier, the performance of a recognition system highly depends on the determination of extracted electrocardiography features. Time domain features, frequency domain features, and statistical measures features for six fundamental waves (PQRSTU) had been used in feature extraction process [14]. Time domain features like morphological features include shapes, amplitudes, and durations were adapted primarily in [15]–[17], frequency domain features like wavelet transformation were widely used [18], [19] stationary features like higher-order statistics also had been developed. Principal component analysis [20] and Hermite functions [21] have been used in electrocardiography classification and related analysis technologies as well. Almost every single published paper proposes a new set of features to be used, or a new combination of the existing ones [1].

The results from these algorithms or models were not amenable to expert labelling, as well as for the identification of complex relationships between subjects and clinical conditions [22]. But for the ambulatory electrocardiography clinical application, as well as the normal application in daily healthcare monitoring for cardiac function or early warning of heart disease, an automated algorithm or model would have significant meaning. The application of artificial intelligence methods has become an important trend in electrocardiography for the recognition and classification of different arrhythmia types [22]. The data explosion puts forward the new request to the method of data processing and information mining.

Over the past decades computational techniques proliferated in the pattern recognition field, simultaneously the applications in electrocardiography recognition, detection and classification for relevant trends, patterns, and outliers. Most of the literatures in the electrocardiography classification task were focused on the supervised learning methods, as in unsupervised learning methods were infrequently used, which needs a lot of effort in labelling data. The MIT-BIH database [23] was the most widely used data in the classification and detection algorithm developments, while mass unlabelled electrocardiography data had been ignored due to the supervise learning approaches essential. Unsuper-

vised learning methods become crucial in mining or analysing unlabelled data, as the unlabelled electrocardiography data accumulated. Unsupervised learning-based approaches and the application to electrocardiogram classification in literatures mainly include clustering-based techniques [21], [24], [25], self-adaptive neural network-based methods [26], [27] and some hybrid unsupervised learning systems [28].

In this paper, we proposed an approach of stacked autoencoder based deep neural network in the unlabelled ambulatory electrocardiography dataset to learn features automatically, with which the cardiac arrhythmia with electrocardiograms classification task was proposed.

In the following sections, we will first introduce the method and classification task in Section II and Section III. In Section IV the classification task are proposed, the experiments results would be illustrated in Section V. Then the discussion are given in Section VI.

II. DEEP LEARNING, AE AND SAE

A. Deep Learning Methods

The backpropagation neural network architecture had been widely applied since 1989 by its multidimensional mapping ability: any L_2 function from $[0, 1]^n$ to \mathbf{R}^n can be implemented to any desired degree of accuracy with a three-layer backpropagation neural network [29]. Until 2006, deep architectures have not been discussed much in the machine learning literature, because of poor training and generalisation errors obtained using the standard random initialization of the parameters [30]. Great successes in speech recognition [31], image recognition [32] had been accomplished via this powerful structure due to the proposed proper training algorithms by Hinton [33]. Deep learning methods attempt to learn feature hierarchies as higher-level features are formed by the composition of lower-level features. The network structure could be first layer-wise initialized via unsupervised training and then tuned with supervised learning methods. Deep models can generate more abstract features at higher levels than the lower ones, better results could be achieved when pre-training each layer with an unsupervised learning algorithm, one layer after the other (the so layer-wised manner), starting with the first layer (that directly takes in input the observed \mathbf{x}) [30].

Different kinds of deep neural network architectures were proposed since the layer-wised training method developed by Hinton. Typical structures include deep belief networks (DBNs) [34], deep Boltzmann machines (DBMs) [35], stacked autoencoders (SAEs) [36], and stacked denoising AEs [37]. The autoencoder based deep learning model and stacked autoencoder as the corresponding architecture.

B. Autoencoders

The first research on the potential benefits of unsupervised learning based pre-training might date back to 1987, in which the first unsupervised autoencoder hierarchies were proposed [38]. The lowest-level autoencoder neural network is a single hidden layer that is trained to map input patterns to themselves. One visible layer of n inputs, one hidden layer of m units and one reconstruction layer of n outputs, as well the

activation functions consist the one-hidden-layer autoencoder as illustrated in (a) part of Fig. 1.

In the training process, the input were mapped to the hidden layer and produce the hidden layer output $h \in \mathbf{R}^m$ which was called “encoder”. The hidden layer outputs were mapped to the output layer that were the reconstruction of input layer as the right part called “decoder” (Fig. 1). An autoencoder is trained to encode the input $x \in \mathbf{R}^n$ into some representations $h \in \mathbf{R}^m$ so that the input can be reconstructed from that representation [30], the reconstructed values were denoted as $\hat{x} \in \mathbf{R}^n$. Mathematically, these two steps can be formulated as

$$h = f(W_1x + b_1) \quad (1)$$

$$\hat{x} = f(\hat{W}_1 + \hat{b}_1) \quad (2)$$

where W_1 and \hat{W}_1 denote the input-to-hidden and the hidden-to-output weights, b_1 and \hat{b}_1 denote the biases. $f(\cdot)$ denotes the activation function that could be a sigmoid function, hyperbolic tangent, and rectified linear function. The autoencoder behaves differently from PCA, which can capture multi-modal aspects of the input distribution (the representation of the input) [39] due to the applying of these nonlinear activation functions.

Here we use W and b as the general parameters of the weights and biases, we use $\hat{x} = h_{W,b}$ denote the reconstruction of the input with an autoencoder, the goal of training is to minimize the error between the input x and reconstruction, i.e.,

$$\operatorname{argmin}_{W,b} \operatorname{error}(x, \hat{x}) \quad (3)$$

So the parameters were W and b while the input x given. The error function can be defined in different ways like root mean square etc. Then the general way for stochastic gradient descent to optimize the parameters were:

$$W := W - \alpha \frac{\partial \operatorname{error}(x, \hat{x})}{\partial W} \quad (4)$$

and

$$b := b - \alpha \frac{\partial \operatorname{error}(x, \hat{x})}{\partial b} \quad (5)$$

in which α stands for the learning rate.

After the training process, the “decoder” part would be removed while the learning features (output of the hidden layer) with weights and biases would be stored, which can be subsequently used for higher layers to produce deeper features.

C. Stacked autoencoder

Stacking the input and hidden layers of AEs together layer by layer constructs an SAE [40], the reconstruction of the input with the stacking method is illustrated in (b) of Fig. 1. Deeper features could be generated from the raw input which could be adopted in a subsequent classifier.

For stacked autoencoder, we first train the first layer autoencoder, subsequently the outputs of the previous layers would be used as the input. Train the parameters of each layer individually while freezing parameters for the remainder of

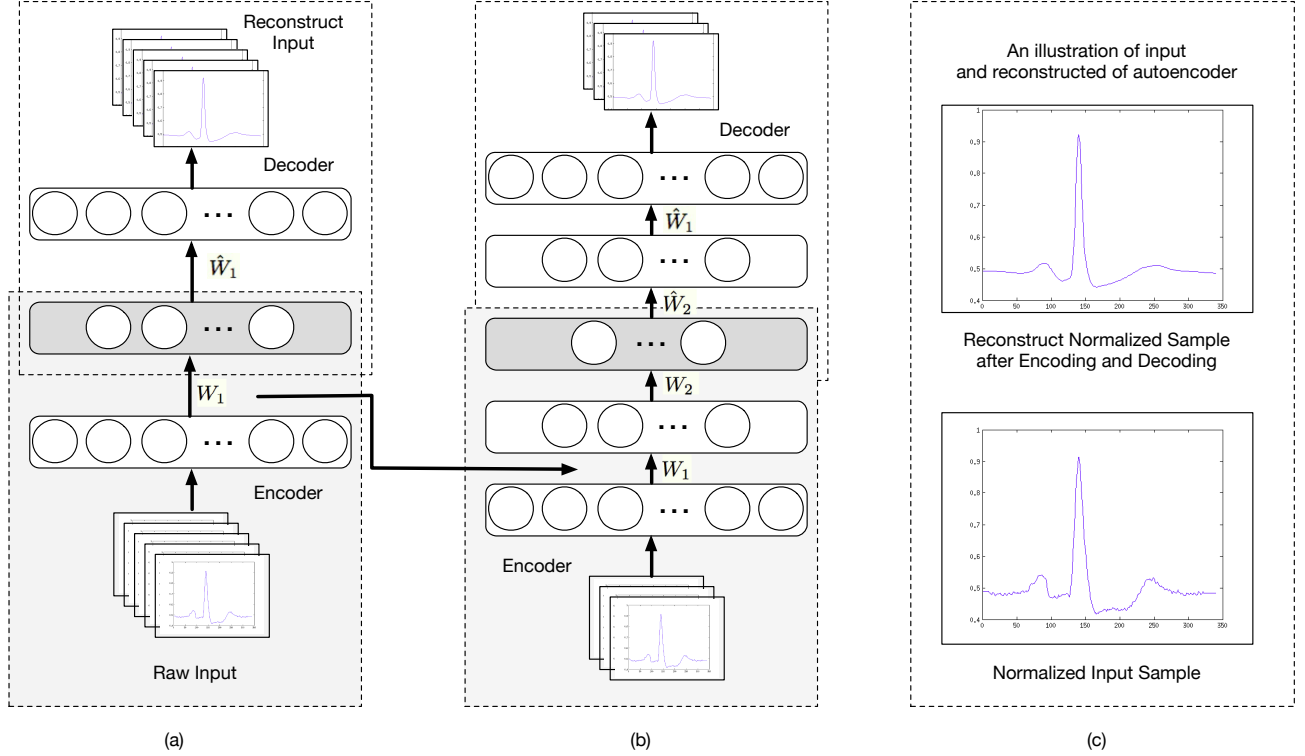


Fig. 1. The autoencoder and the stacked autoencoder. (a) the one-layer autoencoder structure with the encoder and decoder for reconstruction of the raw input; (b) the multi-layer stacked autoencoder for input reconstruction; (c) an illustration of the input sample with the reconstruction.

the model. This parameter training method was called greedy layer-wise training [36].

III. ELECTROCARDIOGRAPHY ANNOTATION

The heart is comprised of the myocardium that rhythmically contract and thus drive the circulation of blood throughout the human body. A wave of electrical current passes through the entire heart, which triggers myocardial contraction [22]. Electrical propagation spreads over the whole heart in a coordinated pattern generate changes on the body surface potentials which can be measured and illustrated as an electrocardiogram (ECG, or sometimes EKG). Metabolic abnormalities (a lack of oxygen, or ischemia, etc.) and pathological changes of the heart engender a variety of ECG. Consequently, ECG analysis has been a routine part of any complete medical evaluation or healthcare applications.

Lots of ECG annotation and diagnosis classification techniques had been proposed in industrial circles and academic communities. As the general steps in a classification problem in a machine learning task, the ECG classification includes data collection, preprocessing, feature extraction, and classification with a classifier. Most of the literatures described models that were combined by different classifier with features extracted by different feature extraction algorithms. The ECG classification methods develop at the same pace with the development of classification theories in machine learning and pattern recognition. Due to the particularity in medical data collection and the medical background requirements for

data annotation, the developments in ECG classification and detection were not as flourishing as the similar research topics like speech recognition, natural language processing, image recognition, etc. The framework of a classification problem that illustrated in Fig. 2.

A. Data Collection

Acquiring and storing ECG data were the base for an analyzing task. Errors might creep into an analysis at any possible stage. Thus, not only the acquisition hardware system but also the transmission and storage should be carefully designed. As for the signal acquiring process, different kinds of sample rates might be involved, for common ECG acquisition device the sample rate would be 128Hz, 250Hz, 340Hz or 500Hz even higher. In murine studies, a sampling rate as high as 2kHz is considered sufficiently. Arbitrary resizing would be an ideal procedure to handle with the different sampling rate from a different data source to build the datasets for analysis, which would be adopted in the experiment to keep data consistency.

B. Data Pre-processing

Before the segmentation and feature extraction process, the ECG signals were pre-processed. As in the procedure of collecting ECG signals, in addition to the ECG signals, the baseline wander (caused by Perspiration, respiration and body movements), power line interference and muscle noise were recorded as well, which had been described in lots of

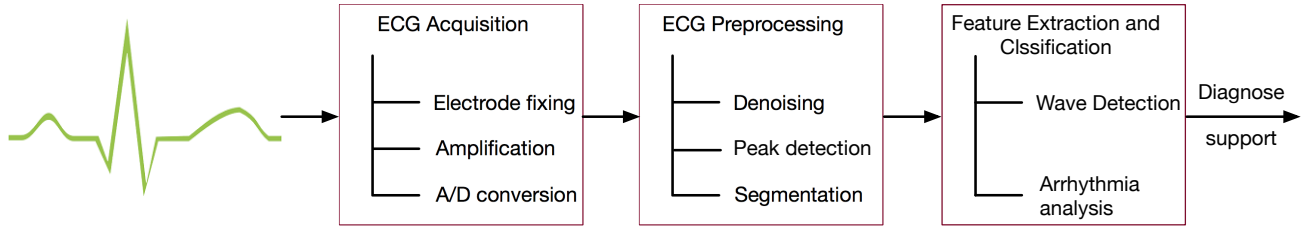


Fig. 2. The technology roadmap of an ECG classification task.

literature [41]. When the filtering methods were proposed and adopted in the preprocessing, the desired information should not be altered. The ECG typically exhibits persistent features like P-QRS-T morphology and average RR interval, and non-stationary features like individual RR and QT intervals, long-term heart rate trends [22]. Possible distortions caused by filtering should be quantified in these features.

The filtered ECG signals then were segmented into individual heartbeat waveforms depends on the detected R peaks in a classification task. The ECG segmentation can be seen as the decoding procedure of an observation sequence regarding beat waveforms [42]. Dynamic time warping [43], time warping [44], Bayesian framework [45], hidden Markov models [42], weighted diagnostic distortion [46], morphology and heartbeat interval based methods [47] and genetic methods [48] had been used in this subtask. The state accuracies were close to 100%, which would be accurate enough in most online and offline applications.

C. Feature Extraction

For supervised learning, the general structure of a learning system includes features and classifiers. For the feature part, time domain features like morphological features include shapes, amplitudes, and durations [15]–[17], frequency domain features like wavelet transformation [18], [19] stationary features like higher order statistics were widely used. With these features, lots of classifier solutions have been proposed for the automated systems to annotate the ECG data [7], [43], [45]. Linear discriminate systems [49], decision tree based methods [50], [51], multilayer perceptron based methods, fuzzy or neuro-fuzzy systems [52], [53], support vector machines classifiers [54] and as well the hybrid systems [55], [56] were adopted extensively.

D. ECG Arrhythmia

After the segmentation for the ECG records, we got plenty of ECG waveform samples with variety categories. Since different physiological disorder may be reflected on the different type of abnormal heartbeat rhythms. For the task of classification, it is quite important to determine the classes would be used. In the early literature, there were no unified class labels for an ECG classification problem. The MIT-BIH Arrhythmia Database was the first available set of standard test material for evaluation of arrhythmia detectors; it played an important role in stimulating manufacturers of arrhythmia analyzers to

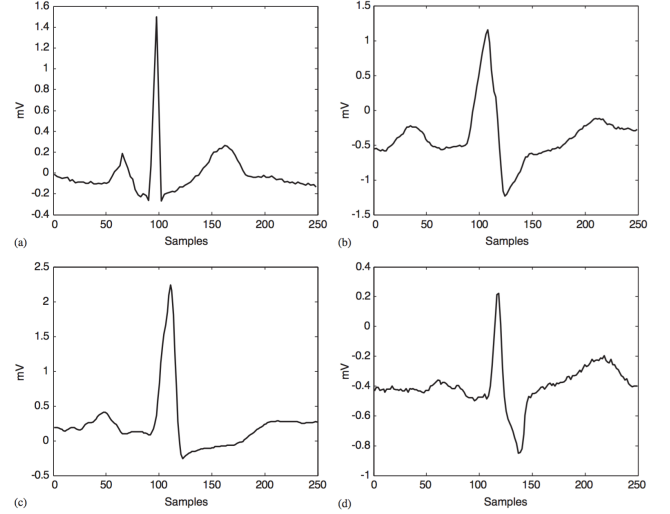


Fig. 3. Waveforms of the ECG beats: (a) Typical N label for normal beat; (b) Typical S label for congestive heart failure beat; (c) Typical V label for Ventricular tachyarrhythmia beat; (d) Typical V label of atrial fibrillation beat.

compete by objectively measurable performance. The annotations in the open database for the ECG categories adopted the ANSI/AAMI EC57: 1998/(R)2008 standard AAMI (2008), which recommended to group the heartbeats into five classes: on-ectopic beats (N as the Fig. 3 (a)); supraventricular ectopic beat (S as the Fig. 3 (b)); ventricular ectopic beat (V as the Fig. 3 (c) and (d)); fusion of a V and a N (F); unknown beat type (Q). These classes or labels have been widely used in the ECG classification tasks related literature. The normal beat, supraventricular ectopic beat and the ventricular ectopic beat categories were used much more frequently while the unknown beat type were abandoned because of its clinical valueless.

IV. CLASSIFICATION WITH DEEP LEARNING STACKED AUTOENCODER STRUCTURES

Since the SAE based deep structure was capable of extracting representations for the raw inputs, it is simpler and more convenient to extract the features for ECG classification instead of discovery or create features by researchers. Though it should be mentioned that for clinical reasons some features from the researchers might be great of importance, the features were limited in the morphology aspect, which lacks the ability to use some importance features in the transform-domain features or statistical representations. Another motivation for

TABLE I
AAMI CLASSES MAPPED FROM MIT-BIH ARRHYTHMIA & LONG-TERM DATABASE TYPES

AAMI heartbeat classes Description	N Any heartbeat not in the S,V,F or Q class	S Supraventricular ectopic beat	V Ventricular ectopic beat	F Fusion beat	Q Unknown beat
MIT-BIH heartbeat types (codes)	Normal beat (1) Left bundle branch block beat (2) Right bundle branch block beat (3) Nodal escape beat (11) Atrial escape beat (34)	Aberrated atrial premature beat (11) Nodal (junctional) premature beat (7) Atrial premature contraction (8) Premature or ectopic supraventricular beat(9)	Ventricular escape beat(10) Premature ventricular contraction (5) Ventricular flutter wave (31)	Fusion of ventricular & normal beat (6)	Paced beat (12) Unclassifiable beat (13) Fusion of paced and normal (38)
MITBIH-AR ^a (100,687)	89,925 ^b	2,774	7,171	802	15
MITBIH-LT(667,347)	600,197	150	64,090	2,906	0

^a Recording 102, 104, 107, 217 were removed.

^b The counts listed in the table may differ from other literatures [2] due to the computation need.

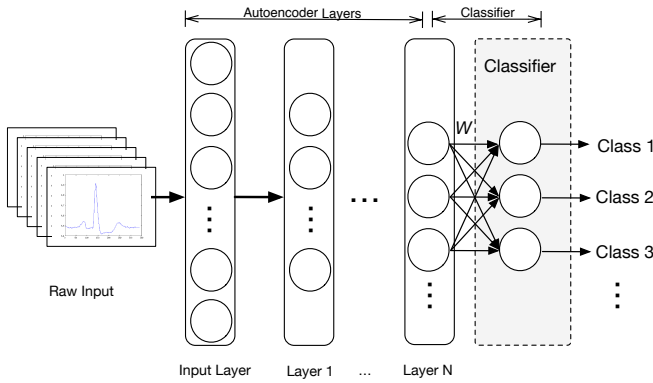


Fig. 4. The multi-layer autoencoder with a subsequent classifier.

the deep features was, in the real use of the long-term ECG monitoring, the variances of the data collection devices and the difference in different objects. For example, different Holter manufactories use different collecting chips, or different analog-to-digital sampling rate, or even different digital signal filter parameters. Another factor is the physiological differences of the objects. According to these factors, using the single morphology or time domain features were insufficient, the probability distribution of a certain class is non-exclusive and has variations over multiple directions in the feature space, which makes it impossible to analyze the waveform point by point in the real complicated situation. More robust and invariant features should be extracted and used in the auto-analysis scenes. It is believed that deep architectures can potentially lead to progressively more abstract features at higher layers of the feature, and more abstract features are invariant to most local changes of the input [57].

To tackle with these problems, with the stacked autoencoder model the deep invariant features of raw ECG samples can be learned layer by layer. Same as the traditional learning systems, after the feature extracting process, a classifier would be constructed behind the neural network to finish the classification task (Fig. 4).

A. Hierarchical Pre-training

1) *Sparsity*: In order to guarantee the representation expression ability of the model, a sparsity constraint is imposed on the hidden units. Let $a_j^{(l)}$ denotes the activation (output of the activation function $f(\cdot)$, in the input layer, $a_i^{(1)} = x_i$), then

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(2)}(x^{(i)})] \quad (6)$$

denotes the average activation of hidden unit j (averaged over the training set). Approximately enforce the constraint:

$$\hat{\rho}_j = \rho \quad (7)$$

where ρ is a sparsity parameter, typically a small value close to zero (such as $\rho = 0.05$), which means the average activation of each hidden neuron j to be close to zero (0.05 for instance). To satisfy the constraint of sparsity, an extra penalty term to the optimization objective that penalized $\hat{\rho}_j$ deviating significantly from ρ . The Kullback-Leibler (KL) divergence:

$$\sum_{j=1}^{s_2} KL(\rho || \hat{\rho}) = \sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \quad (8)$$

is chosen as the penalty term. KL-divergence is a standard function for measuring how different two different distributions are.

2) *Cost Function*: The first step is to learn a deep feature for the ECG samples via pretraining the SAE hierarchically. In Section II, the outline of training method had been derived. Mathematically, the overall cost function of neural network is denoted by $J(W, b)$ which was defined by:

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} W_{ji}^{(l)2} \quad (9)$$

in the first part of which is an average sum-of-square error term: x is the input as the training examples; y denotes the output values (in the autoencoder case, y was set as $y = x$);

$(x^{(i)}, y^{(i)})$ denotes the i -th training example; $W_{ij}^{(l)}$ denotes the weights of the connections between unit j in layer l , and unit i in layer $l+1$; $b_i^{(l)}$ denotes the bias term associated with unit i in layer $l+1$. For the regularization term that tends to decrease the magnitude of the weights and helps prevent of overfitting: λ was adopted as the weight decay parameter while n_l stands for the number of layers, s_l denotes the number of units in layer l . Add the sparsity parameters, in the autoencoder neural network training, the cost function of $J_{sparse}(W, b)$ was defined as:

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho || \hat{\rho}_j) \quad (10)$$

β denotes the weight of the sparsity penalty term.

3) *Activation Function*: The nonlinear mapping activation function $f(\cdot)$ is set to be a *sigmoid* function:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

both in the “encoder” and “decoder”. While training the autoencoder, the “tied weights” are used with the greedy layer-wise approach.

4) *Training*: The stacked autoencoder consists multiple layers of sparse autoencoders in which the outputs of each layer is wired to the inputs of the successive layer. Sequently we use the definitions above, let $W^{(k)}, \hat{W}^{(k)}, b^{(k)}, \hat{b}^{(k)}$, denote the parameters for the k -th autoencoder. The encoding step for the stacked autoencoder is given by running the encoding step of each layer forwardly:

$$a^{(l)} = f(z^{(l)}) \quad (12)$$

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} \quad (13)$$

The decoding step is given by running the decoding stack of each autoencoder in reverse order (in the reconstruction encoding and decoding structure):

$$a^{(n+l)} = f(z^{(n+l)}) \quad (14)$$

$$z^{(n+l+1)} = \hat{W}^{(n-l)} a^{(n+l)} + \hat{b}^{(n-l)} \quad (15)$$

Our goal is to minimize the cost function $J_{sparse}(W, b)$ (with the sparse penalty term) as the function of W and b . To train the network,

After training the network, the reconstruction layers are removed, then stacked autoencoder is constructed with encoders layer by layer. The learned features which we interested in are contained within $a(n)$, which is the activations of the deepest layer of hidden units.

The vector gives a representation of the input that can be used for classification problems by feeding $a(n)$ to a classifier. First initialize the parameters of $W_{ij}^{(l)}$ and each $b_i^{(l)}$ to random values near zero, and then apply an optimization algorithm such as mini-batch gradient descent. Then the iteration of gradient descent updates of W and b were:

$$W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J_{sparse}(W, b) \quad (16)$$

and

$$b_i^{(l)} := b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J_{sparse}(W, b) \quad (17)$$

where α is the learning rate.

Given the training example (x, y) (here in autoencoder (x, \hat{x})), we first run forward to compute the activations of the network, as the output values were the results of hypothesis $h_{W,b}(x)$ as in equation (9). For each node i in layer l , let $\delta_i^{(l)}$ denotes the error term, then the output layer is $\delta_i^{(n_l)}$ means the difference between network's activation and y (\hat{x} in autoencoder) and the hidden layer $\delta_i^{(l)}$ based on the weighted average of the error terms of the nodes that uses $a_i^{(l)}$ as an input.

Algorithm 1 The Backpropagation Algorithm

- 1: Random initialize the parameters and adopt the feedforward calculation to compute the activations for layers L_2 , L_3 to the output layer L_{n_l} .

- 2: For the output layer unit i ,

$$\begin{aligned} \delta_i^{(n_l)} &= \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 \\ &= -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)}) \end{aligned} \quad (18)$$

- 3: For the hidden layer l ,

$$\begin{aligned} \delta_i^{(l)} &= \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)}) \\ &\quad + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1 - \rho}{1 - \hat{\rho}_i} \right) f'(z_i^{(l)}) \end{aligned} \quad (19)$$

- 4: Compute the partial derivatives:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)} \quad (20)$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)} \quad (21)$$

- 5: Update the parameters:

$$\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \alpha \left[\left(\frac{1}{m} \Delta \mathbf{W}^{(l)} \right) + \lambda \mathbf{W}^{(l)} \right] \quad (22)$$

$$\mathbf{b}^{(l)} = \mathbf{b}^{(l)} - \alpha \left[\frac{1}{m} \Delta \mathbf{b}^{(l)} \right] \quad (23)$$

B. Fine-tuning and Classification

To utilize the learned feature integrate the networks structure, we need to fine-tune the pre-trained network, which uses softmax as the output-layer activation.

1) *Softmax model*: The Softmax ensures the activation of each output unit sums to 1 as the outputs could be considered as a set of conditional probabilities. Softmax Regression is a supervised learning algorithm. We use Softmax regression as the fine tuning treats all layers of a stacked autoencoder as one single model, which we can improve all the weight parameters in one iteration.

In the softmax regression model, the hypothesis function is set as:

$$h_{\theta}(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} \quad (24)$$

$$= \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \quad (25)$$

in which output a k dimensional vector denotes k estimated probabilities which stand for k different possible value of the probability. $\theta_1, \theta_2, \dots, \theta_k \in \mathbb{R}^{n+1}$ are the parameters of the model. The term $\frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}}$ normalizes the distribution which sums to one. The cost function in softmax supervised learning is:

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^k \sum_{j=0}^n \theta_{ij}^2 \quad (26)$$

in which $1\{\cdot\}$ is the indicator function. Use the iterative optimization algorithm such as gradient descent method. Taking derivatives, the gradient is:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m [x^{(i)} (1\{y^{(i)} = j\} - p(y^{(i)} = j|x^{(i)}; \theta))] \quad (27)$$

in which $\nabla_{\theta_j} J(\theta)$ is a vector, so that its l -th element is $\frac{\partial J(\theta)}{\partial \theta_{jl}}$ the partial derivative of $J(\theta)$ with respect to the l -th element of θ_j . Perform the update rule $\theta_j := \theta_j - \alpha \nabla_{\theta_j} J(\theta)$ (for each $j = 1, \dots, k$), we can get the parameters.

2) *Fine-tuning with softmax regression*: The output layer size is set to be the same as the usual four categories of arrhythmia, and the input layer has the same size as the stacked autoencoder learned features. Since the Softmax regression is implemented as a single-layer neural network, it can be merged with the former layers of networks to get a deep classifier. The fine-tuning step would use the whole combined architecture for optimization with the backpropagation algorithm as Algorithm 1 illustrated, which has a small learning rate on the former autoencoder layers due to the unsupervised learning process.

V. EXPERIMENTAL RESULTS

In this study, the method adopted is quite different from other ECG classification methods, which extract some features from the segmented ECG samples. This deep network based framework use all the sample points in the ECG samples and let the autoencoder learn the feature by itself. Traditional systems adopted includes the feature extraction part to extract features by artificial design.

Algorithm 2 Stacked Autoencoder with Softmax Regression

```

1: begin
2: initialize training batch size  $a$ , pretraining learning rate  $\alpha$ ,
   numbers of layers  $n_l$ , numbers of neurons in each hidden
   layer  $s_l$ , input dimension  $X$ , classification labels  $k$ .
3: for each layer ( $1 \leq l \leq n_l$ ) do
4:   Construct an autoencoder with input neurons, hidden
   neurons.
5:   if  $l$  is the first layer (i.e.,  $l=1$ ) then
6:     AE_in =  $X$ 
7:     AE_hid =  $h[1]$ 
8:     Set input of the autoencoder to be the raw ECG
   data sample
9:   else
10:    AE_in =  $h[L-1]$ 
11:    AE_hid =  $h[L]$ 
12:    Set input of the autoencoder to be the output of its
   former layer.
13:   end if
14: end for
15:
16: initialize AE weight matrix W with random variables,
   and biases as zeros.
17: for  $e$  do each pretraining epoch
18:   for  $e$  do every mini-batch
19:     Compute reconstruction (based on (12) - (16)):
         $z = f(\hat{W}f(Wx + b) + \hat{b})$  (28)
20:     Compute the cost function:  $J_{sparse}(W, b)$ 
21:     Update parameters using (22) - (23).
22:   end for
23: end for
24: Remove the reconstruction layer.
25:
26: initialize the softmax regression layer input neurons as
    $h[n_l]$ , output neurons as the labeled class.
27: for one fine-tuning epoch do
28:   for every mini-batch do
29:     Use the cost function in (26) and the iteration
   equation in backpropagation algorithm.
30:   end for
31: end for
32: end

```

A. Data Source

Data from the ambulatory electrocardiography database were used in this study, which includes recordings of 100 subjects with arrhythmia along with normal sinus rhythm. The database contains 100 recordings, each containing a 3-lead 24-hour long electrocardiography which were bandpass filtered at 0.1-100Hz and sampled at 128Hz. In this study, only the lead I data were adapted after preprocessing in the classification task. The reference average heart beats for each sample has 97,855 beats for the 24-hour long recording, and the reference arrhythmia average is 1,810 beats which were estimated by a commercial software (this statistics aim to

indicate the existence for arrhythmia samples, which should not be consider as a experiment preset).

The MIT-BIH Arrhythmia Database [23] contains 48 half-hour recordings each containing two 30-min ECG lead signals (lead A and lead B), sampled at 360Hz. As well only the lead I data were used in the proposed method. In agreement with the AAMI recommended practice, the four recordings with paced beats were removed from the analysis. Five records randomly selected were used to verify the real time application. The remaining recordings were divided into two datasets, with small part of which were used as the training set of the fine-tuning process.

The MIT-BIT Long-term Database is also used in this study for training and verification, which contains 7 long-term ECG recordings (14 to 22 hours each), with manually reviewed beat annotations and sampled at 128Hz. Similarly, the 7 recordings were divided into two datasets, with part used as the fine-tuning training set. A description of the labelled datasets are illustrated in Table II .

TABLE II
SAMPLES AFTER SEGEMENTATION

Ambulatory ECG Database (AECG)	MITBIH-AR	MITBIH-LT
9,785,500	100,687	667,343

B. Data Preprocessing & Sample Segementation

Similar to the routine of electrocardiography classification task, the workflow consists of the stages of prepossessing, processing and the classifying. The prepossessing stage related technologies are not the focus of this study, so the classical methods for prepossessing were adapted, and just a brief introduction of the details would be mentioned.

In the preprocessing stage, filtering algorithms were adapted to remove the artefact signals from the ECG signal. The signals include baseline wander, power line interference, and high-frequency noise. For the unlabelled database of ambulatory ECG and the MITBIH LT database, the Lead I data were extracted and a resample from 128Hz to 360Hz procedure was adopted for data consistency.

Before the segmentation procedure from the long-time monitoring ECG signalsHeartbeat Detection: For the heartbeat detection, the MIT-BIH database and unlabelled database, the positions of R waves are determined. The provided fiducial points of R wave had been used as the basis of wave segmentation. The details of the implementation of R wave detection would not be described in this study, and a reference for the R wave detection algorithms had been explored in [58].

In the heartbeat segmentation process, the segmentation program of Laguna [59] was adapted, which also had been validated by other related work [2]. The segmentation process was focus on the Lead I of the recordings. After the segmentation for the ambulatory ECG database, three parts of heartbeat samples listed in Table II were acquired for the classification task. As for the pretraining, fine-tuning for our proposed task and comparison, we divided all the samples into three groups: the pretraining group as dataset A, the fine-tuning group as

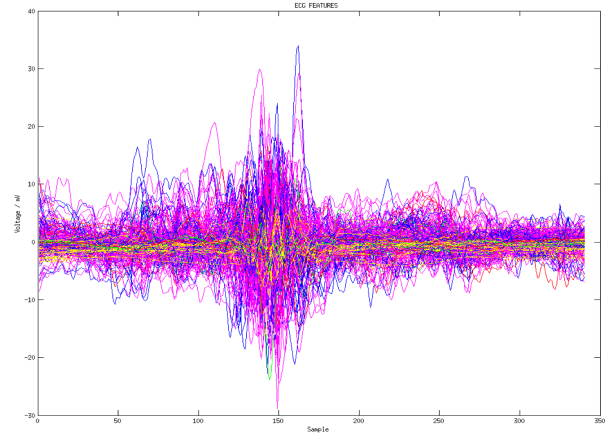


Fig. 5. The overlay chart of features learned by the deep model.

dataset B and test group as dataset C (illustrated in Table III). Samples are chosen randomly from the original AR and LT database, the details of the sample class would be described in the experiment result analysis.

TABLE III
SAMPLES DATASET SETTINGS

Dataset	A	B	C
Useage	Pretraining	Fine-tuning	Test
Source (samples)	AECG (9,785,500)		
	AR (50,193)	AR (33,663)	AR (16,831)
	LT (587,347)	LT (50,000)	LT (30,000)
Total	10,423,040	83,633	46,831

As the algorithm of the deep structure training illustrated, both unsupervised learning and supervised learning are involved in the training process. The pre-training mainly used the unlabelled data to train the autoencoder parts, which only need to set the outputs equal to the inputs. Training data adopted in the unsupervised learning step include the whole samples from the ambulatory electrocardiography database and parts of the MIT-BHI database samples. In the supervised learning step, the MIT-BHI database samples with labels were adopted.

C. Autoencoder Behaviour and Analysis

1) *Reconstruction and the Learned Features:* Single layer AEs are basic building blocks in the model, so we investigate the behavior of one-layer autoencoder. We use a single-layer AE with 100 hidden units. It is shown that the AE do learn a reconstruction of the training sample. We can see that the learned hidden activity retains enough information from the input as part (c) in Fig. 1. Since in the classifier layer, the learned features are

2) *The Classification Test Results:* Test and Classifier Performance Assessment: After the pretraining and fine-tuning process, the deep network parameters were acquired. Then we use the parameters and the test dataset C to predict the class of samples. It is necessary to mention that in A and B, the

labeled data used in pre-training and fine-tuning were divided randomly, which satisfy the requirement of Holdout cross-validation scheme so that the test results were meaningful for the classification task performance improvement. The following statistical parameters of test performance were used in the study:

- 1) Specificity: number of correctly classified normal beats over total number of normal beats.
- 2) Sensitivity: number of correctly classified abnormal beats over total number of the given abnormal beats.
- 3) Overall classification accuracy: number of correctly classified beats over number of total beat.

We adopt a 4-hidden-layer autoencoder deep model for the feature learning for the classification task. Since the labelled data were divided randomly, in the test dataset C, we have 41,886 normal heartbeats, 559 supraventricular ectopic beats, 4,137 ventricular ectopic beats and 244 fusion heartbeats as illustrated in Table IV. Based on the recognition results we got an accuracy of 99.34%. For the N class the specificity is 99.74%, the sensitivity of S class is 82.29%, the sensitivity of V class is 98.31%, the sensitivity of F class is 87.71%.

TABLE IV
TEST RESULT FOR A 4-HIDDEN-LAYER DEEP NETWORK

		Algorithm label					
		N	S	V	F	Q	T
Reference label	N	41,778	38	48	17	5	41,886
	S	93	460	3	1	2	559
	V	52	1	4,067	11	6	4,137
	F	15	0	13	214	2	244
	Q	1	0	1	1	1	5

The test accuracy is about 99.34%.

D. Comparisons with Other Work

In this study, we proposed a deep learning model based approach for the heartbeat arrhythmia classification, the arrhythmia classification problem had been discussed in lots of literature as the reference listed. Different kinds of performance assessment criteria had been adopted. In the comparison part, we adopt several ordinary indicators for the performance assessment, which brought in the above sections. The accuracies, N-class specificities (N-spe), S-class sensitivities (S-sen), V-class sensitivities (V-sen) and the F-class sensitivities (F-sen) in Table V are presented for the comparison. The percentages are calculated from the literature's test results, in which some of the classes are ignored like [3], we use a * symbol to represent the result are not available. In Table V, we use the highest value (2 to 4 hidden layers based structures) for the verification which illustrated in "proposed" line.

Through the comparisons in Table V, we can see that the proposed method offers better accuracy of the classification problem. Since accuracies in lots of the literature were good enough, the verification parameter depends on mainly on the normal class detection, but on the contrary with these methods, this approach provided better performance than other kinds of arrhythmia waveforms classes. Especially in the ventricular ectopic beat sensitivity, a quite large improvement had been made by the proposed method.

TABLE V
COMPARISONS WITH OTHER WORK

Approaches	Accuracy	N-spe	S-sen	V-sen	F-sen
Proposed	99.34%	99.76%	82.29%	98.31%	87.71%
Mar [1]	84.63%	84.85%	82.90%	86.72%	51.55%
Chazal [2]	86.19 %	86.86%	83.83%	77.74%	89.43%
Melgani [3]	90.52%	89.12%	* ^a	89.97%	*
Jiang [4]	94.51%	98.73%	50.59%	86.61%	35.78%

^a * means the results were not available.

^b The listed percentages are based on the previous described rules.

VI. DISCUSSION AND CONCLUSIONS

The deep neural network structure and deep learning algorithms had been widely used in modern computing science especially in Big Data processes, this study proposed one possibility to adopt this method in the health informatics Big Data applications. In this paper, a stacked autoencoder based deep model was proposed for the electrocardiography arrhythmia classification task. It is shown that the features extracted by autoencoder structure are useful in arrhythmia analysis, even increase the accuracy and some abnormalities specification compare with other features like the morphology features or frequency features.

For ECG data arrhythmia classification, the proposed SAE-SR method has been proven to provide statistically better performance than several state-of-art according to the AAMI standard generally. The classification accuracy is 99.34%, for normal heartbeat the specificity attains 99.76%, for supraventricular ectopic heartbeat the sensitivity reach 82.29%, for ventricular ectopic heartbeat the sensitivity achieve 98.31% and 87.71% for fusion heartbeat. Especially in the abnormality type of ventricular ectopic heartbeat, the performance is quite a large improvement. Since the SAE model learned features are based on the unlabelled datasets with low percentage for the abnormalities due to the physiological property, the performance could be improved with more training data. Even though the disadvantage of the SAE-SR method is the training time, but the potential of better results for classification and the testing time efficiency are superior to other methods.

ACKNOWLEDGMENT

The ambulatory ECG database was provided by the Biomedical Instruments Co.,Ltd. This study was financed partially by the National 863 Program of China (Grant No. 2012AA02A604), the Next generation communication technology Major project of National S&T (Grant No. 2013ZX03005013), the Key Research Program of the Chinese Academy of Sciences, and the Guangdong Innovation Research Team Funds for Image-Guided Therapy and Low-cost Healthcare.

REFERENCES

- [1] T. Mar, S. Zaunseder, J. Martinez, M. Llamado, and R. Poll, "Optimization of ecg classification by means of feature selection," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 8, pp. 2168–2177, Aug 2011.

- [2] P. de Chazal, M. O'Dwyer, and R. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.
- [3] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 5, pp. 667–677, Sept 2008.
- [4] W. Jiang and S. Kong, "Block-based neural networks for personalized ecg signal classification," *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1750–1761, 2007.
- [5] T. Olmez, "Classification of ecg waveforms by using rce neural network and genetic algorithms," *Electronics Letters*, vol. 33, no. 18, pp. 1561–1562, Aug 1997.
- [6] C.-W. Lin, Y.-T. Yang, J.-S. Wang, and Y.-C. Yang, "A wearable sensor module with a neural-network-based activity classification algorithm for daily energy expenditure estimation," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 5, pp. 991–998, Sept 2012.
- [7] S. Osowski and T. H. Linh, "Ecg beat recognition using fuzzy hybrid neural network," *Biomedical Engineering, IEEE Transactions on*, vol. 48, no. 11, pp. 1265–1271, Nov 2001.
- [8] M. Kundu, M. Nasipuri, and D. Basu, "A knowledge-based approach to ecg interpretation using fuzzy logic," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 2, pp. 237–243, Apr 1998.
- [9] R. Andreao, B. Dorizzi, and J. Boudy, "Ecg signal analysis through hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 8, pp. 1541–1549, 2006.
- [10] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 37, no. 9, pp. 826–836, 1990.
- [11] Y. Zhu, A. Shayan, W. Zhang, T. L. Chen, T.-P. Jung, J.-R. Duann, S. Makeig, and C.-K. Cheng, "Analyzing high-density ecg signals using ica," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 11, pp. 2528–2537, Nov 2008.
- [12] A. Kampouraki, G. Manis, and C. Nikou, "Heartbeat time series classification with support vector machines," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 4, pp. 512–518, 2009.
- [13] A. Khandoker, M. Palaniswami, and C. Karmakar, "Support vector machines for automated recognition of obstructive sleep apnea syndrome from ecg recordings," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 1, pp. 37–48, Jan 2009.
- [14] C.-P. Shen, W.-C. Kao, Y.-Y. Yang, M.-C. Hsu, Y.-T. Wu, and F. Lai, "Detection of cardiac arrhythmia in electrocardiograms using adaptive feature extraction and modified support vector machines," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7845 – 7852, 2012.
- [15] I. Jekova, G. Bortolan, and I. Christov, "Assessment and comparison of different methods for heartbeat classification," *Medical Engineering & Physics*, vol. 30, no. 2, pp. 248–257, 2008.
- [16] I. Christov, G. Gomez-Herrero, I. Krasteva, I. Jekova, A. Gotchev, and K. Egiazarian, "Comparative study of morphological and time frequency ecg descriptors for heartbeat classification," *Medical Engineering & Physics*, vol. 28, no. 9, pp. 876–887, 2006.
- [17] C. Ye, B. Kumar, and M. Coimbra, "Heartbeat classification using morphological and dynamic features of ecg signals," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 10, pp. 2930–2941, Oct 2012.
- [18] O. T. Inan, L. Giovannrandi, and G. T. A. Kovacs, "Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2507–2515, 2006.
- [19] S. Banerjee and M. Mitra, "Application of cross wavelet transform for ecg pattern analysis and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 326–333, 2014.
- [20] T. Stamkopoulos, K. Diamantaras, N. Maglaveras, and M. Strintzis, "Ecg analysis using nonlinear pca neural networks for ischemia detection," *Signal Processing, IEEE Transactions on*, vol. 46, no. 11, pp. 3058–3067, Nov 1998.
- [21] M. Lagerholm, C. Peterson, G. Braccini, L. Edenbrandt, and L. Sornmo, "Clustering ecg complexes using hermite functions and self-organizing maps," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 7, pp. 838–848, Jul 2000.
- [22] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods And Tools for ECG Data Analysis*. Norwood, MA, USA: Artech House, Inc., 2006.
- [23] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13).
- [24] H. Nishizawa, T. Obi, M. Yamaguchi, and N. Ohya, "Hierarchical clustering method for extraction of knowledge from a large amount of data," *Optical Review*, vol. 6, no. 4, pp. 302–307, 1999.
- [25] C. Maier, H. Dickhaus, and J. Gittinger, "Unsupervised morphological classification of qrs complexes," in *Computers in Cardiology, 1999, 1999*, pp. 683–686.
- [26] S. Palreddy, W. J. Tompkins, and Y. H. Hu, "Customization of ecg beat classifiers developed using som and lvq," in *Engineering in Medicine and Biology Society, 1995., IEEE 17th Annual Conference*, vol. 1, Sep 1995, pp. 813–814 vol.1.
- [27] M. Risk, J. Sobh, and J. Saul, "Beat detection and classification of ecg using self organizing maps," in *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, vol. 1, Oct 1997, pp. 89–91 vol.1.
- [28] P. Tadejko and W. Rakowski, "Hybrid wavelet-mathematical morphology feature extraction for heartbeat classification," in *EUROCON, 2007. The International Conference on Computer as a Tool*, Sept 2007, pp. 127–132.
- [29] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks, 1989. IJCNN., International Joint Conference on*, 1989, pp. 593–605 vol.1.
- [30] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.
- [31] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
- [32] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.
- [33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
- [34] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [35] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.
- [36] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.
- [37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
- [38] D. H. Ballard, "Modular learning in neural networks," in *AAAI*, 1987, pp. 279–284.
- [39] N. Japkowicz, S. Jose Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to pca," *Neural Comput.*, vol. 12, no. 3, pp. 531–545, Mar. 2000.
- [40] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 6, pp. 2094–2107, 2014.
- [41] M. Blanco-Velasco, B. Weng, and K. E. Barner, "Ecg signal denoising and baseline wander correction based on the empirical mode decomposition," *Computers in biology and medicine*, vol. 38, no. 1, pp. 1–13, 2008.
- [42] R. V. Andreão, B. Dorizzi, and J. Boudy, "Ecg signal analysis through hidden markov models," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 8, pp. 1541–1549, 2006.
- [43] H. Vullings, M. Verhaegen, and H. Verbruggen, "Automated ecg segmentation with dynamic time warping," in *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*. IEEE, 1998, pp. 163–166.
- [44] H. Vullings, M. Verhaegen, and H. B. Verbruggen, "Ecg segmentation using time-warping," in *Advances in Intelligent Data Analysis Reasoning about Data*. Springer, 1997, pp. 275–285.

- [45] O. Sayadi and M. Shamsollahi, "A model-based bayesian framework for ecg beat segmentation," *Physiological Measurement*, vol. 30, no. 3, p. 335, 2009.
- [46] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (wdd) measure for ecg signal compression," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 11, pp. 1422–1430, 2000.
- [47] P. De Chazal, M. O. Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 7, pp. 1196–1206, 2004.
- [48] A. Gacek and W. Pedrycz, "A genetic segmentation of ecg signals," *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 10, pp. 1203–1208, 2003.
- [49] M. F. Shinwari, N. Ahmed, H. Humayun, I. ul Haq, S. Haider, and A. ul Anam, "Classification algorithm for feature extraction using linear discriminant analysis and cross-correlation on ecg signals," *International Journal of Advanced Science and Technology*, vol. 48, pp. 149–161, 2012.
- [50] V. Krasteva, R. Leber, I. Jekova, R. Schmid, and R. Abacherli, "Classification of supraventricular and ventricular beats by qrs template matching and decision tree," in *Computing in Cardiology Conference (CinC), 2014*. IEEE, 2014, pp. 349–352.
- [51] F. Charfi and A. Kraiem, "Comparative study of ecg classification performance using decision tree algorithms," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 3, no. 4, pp. 102–120, 2012.
- [52] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *International Conference on artificial intelligence and statistics*, 2009, pp. 153–160.
- [53] M. Vafaie, M. Ataei, and H. Koofgar, "Heart diseases prediction based on ecg signals classification using a genetic-fuzzy system and dynamical model of ecg signals," *Biomedical Signal Processing and Control*, vol. 14, pp. 291–296, 2014.
- [54] E. D. Übeyli, "Ecg beats classification using multiclass support vector machines with error correcting output codes," *Digital Signal Processing*, vol. 17, no. 3, pp. 675–684, 2007.
- [55] M. R. Homaeinezhad, S. Atyabi, E. Tavakkoli, H. N. Toosi, A. Ghaffari, and R. Ebrahimpour, "Ecg arrhythmia recognition via a neuro-svm-knn hybrid classifier with virtual qrs image-based geometrical features," *Expert Systems with Applications*, vol. 39, no. 2, pp. 2047–2058, 2012.
- [56] L. Vanitha and G. Suresh, "Hybrid svm classification technique to detect mental stress in human beings using ecg signals," in *Advanced Computing and Communication Systems (ICACCS), 2013 International Conference on*. IEEE, 2013, pp. 1–6.
- [57] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [58] V. Afonso, W. J. Tompkins, T. Nguyen, and S. Luo, "Ecg beat detection using filter banks," *Biomedical Engineering, IEEE Transactions on*, vol. 46, no. 2, pp. 192–202, Feb 1999.
- [59] L. Sörnmo and P. Laguna, "Electrocardiogram (ecg) signal processing," *Wiley Encyclopedia of Biomedical Engineering*, 2006.

PLACE
PHOTO
HERE

Xingbin Qin is pursuing his master degree of computer science at the University of Chinese Academy of Sciences. His current research interests include digital signal processing, machine learning and biomedical engineering.

PLACE
PHOTO
HERE

Jianping Fan received the B.Eng. degree in the Nankai University, and the Ph.D. degree from the Chinese Academy of Sciences. He is now a researcher with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His current research interests include cloud computing, high performance computing.

PLACE
PHOTO
HERE

Lei Wang received the B.Eng. degree in information and control engineering, and the Ph.D. degree in biomedical engineering from Xian Jiaotong University, Xian, China, in 1995 and 2000, respectively. He was with the University of Glasgow, Glasgow, U.K., and Imperial College London, London, U.K., from 2000 to 2008. He is currently a full professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He has published over 200 scientific papers, authored four book chapters, and holds 60 patents. His current research interests include body sensor network, digital signal processing, and biomedical engineering

PLACE
PHOTO
HERE

Yan Yan received the B.eng. degree in the Harbin Institute of Technology, Wei Hai, and the master of engineering in the Harbin Institute of Technology in 2010 and 2012, respectively. He was a research assistance in the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences from 2012 to 2014. Since 2014 he start pursuing his doctor's degree of computer science. His current research interests include digital signal processing, machine learning and biomedical engineering.