Automatic classification of CAD ECG signals with SDAE and bidirectional long short-term term network

ERIC KE WANG¹, XUN ZHANG¹, LEYUN PAN²

¹Harbin Institute of Technology, Shenzhen 518055, China (e-mail: hit@hit.edu.cn(E.K.W.), zhangxun@stu.hit.edu.cn(X.Z.))
²German Cancer Research Center, 69120 Heidelberg, Germany (e-mail: l.pan@dkfz.de(L.Y.P.)

Corresponding author: Eric Ke Wang (e-mail: wk_hit@hit.edu.cn).

This paper is a cooperative research article from HIT China with German Cancer Research Center, which was supported in part by National Natural Science Foundation of China (No.61572157), grant No.2016A030313660 and 2017A030313365 from Guangdong Province Natural Science Foundation, JCYJ20160811155158682 and JCYJ20160428092427867 from Shenzhen Municipal Science and Technology Innovation Project. The authors thank the reviewers for their comments.

ABSTRACT Coronary artery disease (CAD) has been one of main causes of heart diseases globally. The electrocardiogram (ECG) is a widely used diagnostic tool to monitor patients’ heart activities, and medical personnel need to judge whether there are abnormal heartbeats according to captured results. Therefore, it is significant to identify ECG signals accurately and fast. In this paper, a fast and accurate electrocardiogram (ECG) classification system based on deep learning is proposed. In our model, stacked denoising autoencoders (SDAE), as encoder, automatically learns semantic encoding of heartbeats without any complex feature extraction in unsupervised way. Then bidirectional LSTM (Bi-LSTM) classifier achieves classification of heartbeats with semantic encoding. SDAE implements noise-reduction while Bi-LSTM takes full advantage of temporal information in data. At the same time, this method relieves impacts from unbalanced data by employing cost-sensitive loss function. We validate our model on MIT-BIH Arrhythmias Database, SVDB and NSTDB respectively. Compared with state-of-art methods, the final result verify that this newly proposed method not only has high accuracy but also boosts classifying efficiency.

INDEX TERMS Arrhythmia, bidirectional long short-term term network (Bi-LSTM), cost-sensitive learning, denoise, electrocardiogram (ECG), stacked denoising autoencoder (SDAE)

I. INTRODUCTION

NOWADAYS, coronary artery disease (CAD) has been one of main causes of global increase in the fatality rate, and arrhythmia is a disease belonging to cardiovascular. Once people are sick for it, it is significant to continuously detect patients’ heart activities with ECG. Proper classification of arrhythmia is meaningful for patients to get suitable therapy.

So far, numerous algorithms have contributed to automatic heartbeats classification. It is very common to employ machine learning methods to classify ECG. Generally, arrhythmia classification includes three steps. They are pre-processing, feature extraction and classification. Among these steps, feature extraction and design of classifier are major work in related researches. As for handcrafted features used to input, there are various available representations such as morphology [7], temporal information, waveforms and so on. There are some common morphological features: RR interval, P wave, QRS wave, ECG-intervals, QRS interval, PR interval, ST interval, ST level [5] and so on. These features are very intuitive and interpretable. However, to obtain them, researchers need to locate each band of heartbeats precisely, which is difficult to achieve. In addition, most papers employ mathematical transformation such as wavelet transform to get some coefficients representing ECG. On the other hand, the basic principles of wavelet transform is esoteric though it extracts multi-scale features. Among these ways, the simplest way is utilizing points of the segmented ECG curve such as heartbeat as features [32]. These features have high dimensions so researchers applied many algorithms on samples to reduce dimensions, i.e., Principal Component Analysis
(PCA), Independent Components Analysis (ICA) [23], [30], Kernal Principal Component Analysis (KPCA) [14].

Meanwhile, many classical classifying models are employed to solve this problem. The four most common algorithms are Support Vector Machine (SVM) [12], artificial neural networks (ANN) [19], Linear Discriminant (LD) [6], and Reservoir Computing With Logistic Regression (RC) [8]. Apart from these, other methods also are utilized to classify arrhythmia like decision tree [28], nearest neighbors [16], hidden Markov models [4], and hyperbox classifier [3]. In recent years, some researchers begin to use deep learning as a tool to extract features. For instance, multilayer perceptron (MLP) [15], [20], [22], probabilistic neural network (PNN) [29], recurrent neural network (RNN) [2], deep belief networks (DBNs) [11] and Convolutional Neural Networks (CNN) [17] are some applied models.

In reality, features extraction demands on researchers, and sometimes those handcrafted features cannot accurately express electrocardiogram. Compared with machine learning, deep learning has outstanding performance in object recognition, time series data, image classification and so on. Since incidence of cardiovascular diseases (CVDs) increases year by year and CVDs usually are paroxysmal, long-term monitoring heart activity is inevitable, which yields mass data. Deep learning has been proved that it did a very good job on huge data. Apparently, deep learning can save time of features extraction and doesn’t require much related knowledge, which bring about high efficiency. Current researchers usually focus on Convolutional Neural Networks (CNN), whereas there are a few research based on long short-term memory (LSTM) networks. Though LSTM networks have outstanding performances than CNN in most cases, it requires much more time than CNN on same data, which is unpractical. Apart from that, ECG is unbalanced data because it has comparatively few negative samples. Thus, the final classified result might be influenced by unbalanced data.

In the experiment, we propose a new architecture employing cost sensitive loss function, and construct a Bi-LSTM classifier based on SDAE. The final outcome shows the newly proposed method can improve accuracy without manual intervention and save a lot of time compared with state-of-art methods. It also reduces impacts from unbalanced data and makes classifying process more robust by SDAE.

II. DATASETS

A. DATA SOURCES

In the research, we use data from Massachusetts Institute of Technology-Beth Israel Hospital arrhythmia database, which is from PhysioNet [9]. MIT-BIH ECG Database is the most widely used and well-known database in the world. Since 1999, with the support of the National Center for Research Resource of National Institutes of Health, this database was open-source on Internet. MIT-BIH ECG Database consists of many sub-databases. Every sub-database includes some particular types of ECG records. In our research, MIT-BIH Arrhythmia Database, MIT-BIH Supraventricular Arrhythmia Database and MIT-BIH Noise Stress Test Database will be utilized.

1) MIT-BIH Arrhythmia Database

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

2) MIT-BIH Supraventricular Arrhythmia database (SVDB)

This database consists of 78 two-lead recordings of approximately 30 min and samples 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 medical students. It is noteworthy that we interpolate samples from this database since its sampling rate is different from others. By interpolation, SVDB is consistent with other database.

3) MIT-BIH Noise Stress Test Database (NSTDB)

This database includes 12 half-hour ECG recordings and 3 half-hour recordings of noise typical in ambulatory ECG recordings. The noise recordings which also samples at 360Hz, were made by adding calibrated amounts of noise to unpolluted ECG recording from MIT-BIH Arrhythmia Database. Apparently, it is compatible with MIT-BIH Arrhythmia Database without any transformation.

B. EXPERIMENT SETUP

To detect arrhythmia, we ought to select objects being classified at first. Apparently, it is not a wise choice to use recordings as samples directly, since one recording contains too much information and the number of recordings is too small for deep learning methods. For most researches, the object being classified is heartbeat, the easiest data to extract [18]. In one ECG recording, there are various kinds of and numbers of heartbeats. In the related study, QRS wave localization algorithm can approximately locate position of heartbeats. In the experiment, we use location of R peak of the QRS-complex provided by head file in database to do ECG signal segmentation. To obtain type and position of each peak easily, we could seek WFDB (WaveForm Database) software package for help. WFDB software package is a specialized software for PhysioBank data and it includes about 75 WFDB applications for signal processing and automated analysis. Besides, the WAVE software in this package also achieves viewing, annotation, and interactive analysis of waveform data. Taking recordings in MIT-BIH arrhythmia database as example, we observe that one recording has 650000 sampled points and contains 3100 or so heartbeats at most. Therefore,
TABLE 1: Relationship between MIT-BIH Database and AAMI classes

<table>
<thead>
<tr>
<th>AAMI</th>
<th>MIT-BIH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>N.L.R</td>
</tr>
<tr>
<td>Supraventricular ectopic beat (SVEB)</td>
<td>c.j,a,a,j,s</td>
</tr>
<tr>
<td>Ventricular ectopic beat (VEB)</td>
<td>V.E</td>
</tr>
<tr>
<td>Fusion</td>
<td>F</td>
</tr>
<tr>
<td>Unknown beat</td>
<td>J.Q</td>
</tr>
</tbody>
</table>

TABLE 2: Numbers of each heartbeat type in the experiments

<table>
<thead>
<tr>
<th>Database</th>
<th>N</th>
<th>S</th>
<th>V</th>
<th>F</th>
<th>Q</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT-BIH(DS1)</td>
<td>90603</td>
<td>2371</td>
<td>7235</td>
<td>802</td>
<td>8041</td>
<td>109462</td>
</tr>
<tr>
<td>SVDB(DS2)</td>
<td>158153</td>
<td>12116</td>
<td>9745</td>
<td>78</td>
<td>22</td>
<td>180114</td>
</tr>
<tr>
<td>NSTDB(DS3)</td>
<td>22248</td>
<td>2760</td>
<td>576</td>
<td>0</td>
<td>0</td>
<td>25584</td>
</tr>
</tbody>
</table>

sample is assigned with 200 dimensions. Following the Association for the Advancement of Medical Instrumentation AAMI recommended practice, original heartbeat types in MIT-BIH database can be divided into five classes which is shown in Table 1.

In next experiments, ten-fold cross-validation approach is adopted to estimate performance of models. Ninety percentage of samples after ECG segmentation will be used to train and remain samples are used to test. The final performance of models is the average of all ten-fold measures. Table 2 describes numbers of each heartbeat type in the experiments.

### III. METHODOLOGIES

In previous section, we have described how to obtain heartbeats by segmenting original ECG recordings. In the experiment, we firstly defined heartbeat as training data. \( D = \{ x_i, y_i \}_{i=1}^n \) is training set containing \( n \) ECG signals, where \( x_i \) is \( d \)-dimensional ECG signal vector and \( y_i \) is its corresponding label.

### A. DATA PREPROCESS

Before calculating features for classification, we ought to preprocess samples by min-max normalization. Min-max normalization will transform each sample as follows:

\[
x'_i = \frac{x_i - \min_{1 \leq j \leq n} \{ x_j \}}{\max_{1 \leq j \leq n} \{ x_j \} - \min_{1 \leq j \leq n} \{ x_j \}}
\]

(1)

\( x'_i \) will be input of proposed approach. Meanwhile, we do not filter noises and try to preserve raw signals to enhance generalization ability of classifier. In subsequent experiments, DS3 containing noises will be used to test denoising ability of proposed architecture.

### B. STACKED DENOISING AUTOENCODER

Original number of dimension is so large that it might cause dimension disaster. Stacked denoising autoencoder (SDAE) is built to extract concealed meaning in heartbeat samples. Apparently, a SDAE consists of several denoising autoencoders (DAE) which are symmetrical neural network to learn some more abstract features by an unsupervised way. To make sure model is robust, original data will be corrupted with some noises and then decoders will reconstruct data based on these noisy corrupted data, so the model is not sensitive to noisy data. At the same time, the dimension of features can be reduced through the SDAE. These concise features extracted from SDAE is the input of Bi-LSTM classifier.

DAE is component of SDAE and also made of encoding part and decoding part, and Figure 1 shows architecture of DAE. It is constructed by several dense layers and specific parameters of its dense layers will be explained in the following part.

Before the encoding part, targeted signal \( x' \in R^d \) is randomly corrupted and become noisy version \( \tilde{x}_i \). The mapping function from input layer to hidden layer is Relu function, so the representation of hidden layer is:

\[
h_i = f(W^{(e)} \tilde{x}_i + b)
\]

(2)

where \( W^{(e)} \in R^{L \times D} \) is the encoder weight matrix, \( b \in R^L \) is the encoder bias vector and \( h_i \in R^L \) is L-dimensional feature. L is assigned artificially. In the decoding part, the hidden representation can be mapped into a reconstructed signal \( z_i \in R^D \):

\[
z_i = f(W^{(d)} h_i + b')
\]

(3)

where \( W^{(d)} \in R^{D \times L} \) is the decoder weight matrix, \( b' \in R^D \) is the decoder bias. To obtain exact low-dimensional representation of input signal \( x \), we should look for suitable network parameter \( \theta_{DAE} = \{ W^{(e)}, W^{(d)}, b, b' \} \). Aim to this problem, we ought to minimize the following cost function:

\[
E_1(\theta_{DAE}) = \frac{1}{2n} \sum_{i=1}^{n} \| x_i - z_i \|^2 + \gamma_1 \left( \left\| W^{(e)} \right\|^2_F + \left\| W^{(d)} \right\|^2_F \right)
\]

(4)

where \( \gamma_1 \) is weight of second term and \( \| \cdot \| \) is the Frobenius norm. The first term means the reconstruction error over all \( n \) training samples and the later term represents sparsity constraint which avoids over-fitting. Finally, according to above algorithm, output of previous hidden layer is the input of next hidden layer. In the same manner, we can stack multiple hidden layers, and remove reconstruction layers of...
SDAE. By SDAE, underlying features in ECG signals can be detected, and can be fed as input data for next Bi-LSTM classifier. Pre-training applies unsupervised Adam algorithm to learn parameters. Until loss meets requirements, the training process of SDAE parameters in hidden layers completes.

C. BIDIRECTIONAL LSTM CLASSIFIER

Once the pre-training of encoder part completed, bidirectional LSTM will be applied as classifier. The Bidirectional LSTM (Bi-LSTM) consists of forward LSTM and backward LSTM, which are connected with softmax regression layer. LSTM was proposed by Hochreiter and Schmidhuber in 1997 [25], and it can remember long-term dependency. Obviously, ECG signals are usually so long that we consider LSTM to solve this problem. In this task, the performance of Bi-LSTM is better than LSTM and thus Bi-LSTM is employed as main framework, which is validated by experiments. Since Bi-LSTM takes previous and latter features into consideration at same time, it can extract more substantial temporal information. Supposed the output from final hidden representation layer of SDAE is $h_i \in R^L$, the input of Bi-LSTM should be reshaped into $h_i \in R^{L+1}$ so it can meet requirements of Bi-LSTM. After the forward LSTM and the backward LSTM, the final output $o_i$ is a 32 dimension vector.

Because we are dealing with a multiclass classification problem, a softmax regression layer is chose to do multi-classification, which yields a task-specific supervised learning. Then we could fine tune the entire model using backpropagation by minimizing the following loss function:

$$
\ell_q(\theta_m) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} I\{y_i = k\} \log \frac{e^{\theta_m^T x(i)}}{\sum_{j=1}^{K} e^{\theta_m^T x(i)}} + \gamma_2 \|W_{softmax}\|_F^2 + \sum_{h=1}^{H} \|W_h\|_F^2
$$

where first term represents the cross entropy loss for the softmax layer, and the second term is weight decay penalty. $I\{\cdot\}$ is an indicator function that get 1 if the statement is true otherwise it takes 0. Through minimizing the loss function, the vector of parameter $\theta_m = \{W_1, \ldots W_H, b_1, \ldots b_H, W_{softmax}\}$ can be correctly estimated. With known labels of ECG signals, we will fine tune our model with enough epochs. Softmax regression layer will output probabilities of every class for each ECG heartbeat sample $x_i$, and model will assign one class label with highest probability to this ECG sample.

D. COST-SENSITIVE LEARNING

So far, a preliminary classification model has been constructed. However, the distribution of heartbeats is unbalanced when we observe dataset. The number of normal heartbeats is much larger than others, which is class imbalance problem. Normal heartbeats accounts for 80% or higher in dataset. Nowadays, there are two main ways to solve class imbalance problem. They are random sampling and cost-sensitive learning. Random sampling tries to make data balanced by changing distribution of original data. There are some major methods: oversampling, downsampling and SMOTE. Nevertheless, random sampling will cause some problems such as overfitting or missing important information. This approach takes cost-sensitive learning to improve results in algorithm. Cost-sensitive learning provides different weights for different types of samples and previous research also has combined cost-sensitive learning and deep learning. In normal task, weights of all samples usually are same. In this scenario, abnormal heartbeats should have higher priority since final wrong results could be excluded artificially. Put it differently, this method attempts to increase specificity. Correspondingly, sensitivity might decrease due to cost-sensitive learning.

The foundation of cost-sensitive learning is constructing a cost matrix whose values represent punishment of misclassification. Table 3 illustrates definition of cost matrix. $C(i, j)$, $i, j \in \{1, 2, 3, 4, 5\}$ denotes cost when model misclassifies samples in j class as i class.

The loss function is:

$$
\ell(i|x) = \sum_j P(j|x)C(i, j)
$$

where $P(j|x)$ represents probability that the sample is classified as j class, and the probability can be obtained by softmax regression layer. By adding cost-sensitive loss to models, the function curve of the original loss function biases.

In cost matrix, there are some principles. First, $C(i, j)$ equals to 0 when $i$ equals to $j$. Second, $C(i, j)$ is greater than zero when $i$ does not equals to 0. In this paper, we assign misclassification cost depending on heartbeat proportions. Because distribution of heartbeat on each database is different, the specific value in cost matrix should be calculated depending on current database. Values on the diagonal line are 0 and other values can be calculated as following formula:

$$
C(i, j) = \frac{Number of j heartbeat}{Number of i heartbeats}
$$

E. PROPOSED MODEL

To classify heartbeats more efficiently and accurately, we propose a Bi-LSTM classifier based on SDAE combining with cost-sensitive learning. SDAE extract underlying information and Bi-LSTM classifier utilizes these information to classify heartbeats. From perspective of algorithm, cost-sensitive loss function decreases inaccuracy due to unbalanced data. Figure 2 illustrates overall flowchart of this approach. First, What is worthy to notice is that feature

![Table 3: Definition of cost matrix](https://example.com/table3.png)

<table>
<thead>
<tr>
<th>True Class</th>
<th>N</th>
<th>V</th>
<th>S</th>
<th>F</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>C(N, N)</td>
<td>C(N, V)</td>
<td>C(N, S)</td>
<td>C(N, F)</td>
<td>C(N, Q)</td>
</tr>
<tr>
<td>V</td>
<td>C(V, N)</td>
<td>C(V, V)</td>
<td>C(V, S)</td>
<td>C(V, F)</td>
<td>C(V, Q)</td>
</tr>
<tr>
<td>S</td>
<td>C(S, N)</td>
<td>C(S, V)</td>
<td>C(S, S)</td>
<td>C(S, F)</td>
<td>C(S, Q)</td>
</tr>
<tr>
<td>F</td>
<td>C(F, N)</td>
<td>C(F, V)</td>
<td>C(F, S)</td>
<td>C(F, F)</td>
<td>C(F, Q)</td>
</tr>
<tr>
<td>Q</td>
<td>C(Q, N)</td>
<td>C(Q, V)</td>
<td>C(Q, S)</td>
<td>C(Q, F)</td>
<td>C(Q, Q)</td>
</tr>
</tbody>
</table>
classification is a supervised training process, and all weights in the framework will be updated with the help of labeled data. What is more, obtained values of SADEs’ weights in unsupervised training process will be set as initial values of SDAE in supervised training process. Entire training process utilizes Adam optimization algorithm and gets an optimum solution. At last, classified results will be output.

IV. EXPERIMENT AND RESULTS

In our experiments, different neural networks are designed to get better results but some training parameters are common such as epoch numbers (100), batch size (248), final feature number (6), optimizer (Adam optimizer). At the same time, since our model is data driven and end-to-end training, we set learning rate to be 0.005 to avoid unsteady gradients. All experiments are carried out on Linux servers (Intel Core i7-6800K, RAM 8GB, CPU 3.4GHZ, 64 bits, GPU NVIDIA GeForce GTX 1080).

A. PERFORMANCE EVALUATION

For performance evaluation, we use the standard measures: sensitivity ($S_e$), specificity ($S_p$), accuracy ($Acc$) which are based on confusion matrix to analyze results. In addition, we also utilize computing time per heartbeat ($CTPH$) to measure efficiency of models. The smaller computing time per heartbeat of model is, the higher efficiency the model has.

Please note that the unit of computing time is millisecond.

Confusion matrix is a specific table layout that can visualize the performance of an algorithm. Each row of the matrix represents the instances in a predicted class and each column represents the instances in an actual class. Based on this matrix, True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) can be calculated. TP represents correctly classified heartbeats that belong to this class and TN stands for incorrectly classified samples which belong to this class. FP means that ECG samples that do not belong to this class are classified into this category. FN denotes that ECG samples do not belong to this class and also not classified into this category. Table 4 depicts the definition of confusion matrix.

<table>
<thead>
<tr>
<th>Class</th>
<th>n</th>
<th>v</th>
<th>s</th>
<th>f</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>$N_n$</td>
<td>$N_v$</td>
<td>$N_s$</td>
<td>$N_f$</td>
<td>$N_q$</td>
</tr>
<tr>
<td>V</td>
<td>$V_n$</td>
<td>$V_v$</td>
<td>$V_s$</td>
<td>$V_f$</td>
<td>$V_q$</td>
</tr>
<tr>
<td>S</td>
<td>$S_n$</td>
<td>$S_v$</td>
<td>$S_s$</td>
<td>$S_f$</td>
<td>$S_q$</td>
</tr>
<tr>
<td>F</td>
<td>$F_n$</td>
<td>$F_v$</td>
<td>$F_s$</td>
<td>$F_f$</td>
<td>$F_q$</td>
</tr>
<tr>
<td>Q</td>
<td>$Q_n$</td>
<td>$Q_v$</td>
<td>$Q_s$</td>
<td>$Q_f$</td>
<td>$Q_q$</td>
</tr>
</tbody>
</table>

Please note that the unit of computing time is millisecond.
are be calculated by following formulas:

\[ S_e = \frac{TP}{TP + FN} \times 100\% \]  \hspace{1cm} (9)  
\[ S_p = \frac{TN}{TN + FP} \times 100\% \]  \hspace{1cm} (10)  
\[ Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \]  \hspace{1cm} (11)  
\[ CTPH = \frac{\text{RunningTime (millisecond)}}{\text{Numbers of Heartbeats}} \]  \hspace{1cm} (12)  

B. EXPERIMENT 1: GENERATION OF INITIAL MODEL

To validate assigned parameters, this experiment is carried out. The main focus are setting of final features number and selection of post-processing classifier. Models in this experiment don’t employ cost-sensitive loss function.

First, according to previous principles, we build an architecture on the basis of SDAE, Bi-LSTM and softmax regression layers. However, we do not know the exact number of layers in SDAE so we design a series of comparisons to find best final features number of SDAE. Therefore, several SDAE with different hidden layers are constructed. Given that the dimension of sample is 200 and final result has 5 types, we use \{201,100,201\}, \{201,100,50,100,201\}, \{201,100,50,25,50,100,201\} and \{201,100,50,25,12,25,50,100,201\} nodes for each configuration, that is alternating between a sparse and dense representation. Strictly speaking, configuration \{201,100,201\} can not be taken accounted into the range of SDAE, but we want to increase diversity of models. After pre-training SDAE on training set without supervision, we can obtain rough compressed data from final hidden layer of corresponding SDAE. These data will be consequently input into a Bi-LSTM classifier, and average performance over ten-fold cross validation is result for comparison.

By analyzing results on three database, computing time per heartbeat is directly proportional to final features number roughly. Table 5 shows the classified results of Bi-LSTM classifier based on SDAE with different number of hidden features.

From Table 5, by and large, all classifiers get good performance. The SDAE ending with 6 gets best result and the SDAE ending with 12 get second best performance in sensitivity, accuracy and computing time per heartbeats. As for specificity, all classifiers perform roughly same. These results demonstrates SDAE ending with 6 nodes is more powerful. This conclusion also proves theory from Lucie [21]. In that experiment, reduced number of features used for ECG classification achieves similar or better performance compared with the whole features.

Though we have learnt model using the SDAE ending with 6 would get better performances, it is also necessary to do another experiment to explain why Bi-LSTM is required. Figure 3-5 shows the comparison of different post-processing after SDAE ending with 6 over 3 database. \( S_e, S_p, Acc \) are more influential criteria when we select post-processing classifier, so CTPH is not employed as performance measure in this experiment.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Final Features Number</th>
<th>( S_e )%</th>
<th>( S_p )%</th>
<th>Acc%</th>
<th>CTPH (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>6</td>
<td>92.79</td>
<td>99.40</td>
<td>95.28</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>89.55</td>
<td>97.34</td>
<td>92.74</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>83.71</td>
<td>98.25</td>
<td>90.95</td>
<td>4.26</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>81.51</td>
<td>98.13</td>
<td>91.17</td>
<td>8.69</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>80.78</td>
<td>99.20</td>
<td>90.90</td>
<td>17.02</td>
</tr>
<tr>
<td>DS2</td>
<td>6</td>
<td>85.57</td>
<td>99.01</td>
<td>97.08</td>
<td>1.32</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>83.53</td>
<td>98.98</td>
<td>96.97</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>83.03</td>
<td>98.84</td>
<td>96.90</td>
<td>4.54</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>79.83</td>
<td>99.24</td>
<td>96.78</td>
<td>8.92</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>69.95</td>
<td>99.24</td>
<td>95.59</td>
<td>18.03</td>
</tr>
<tr>
<td>DS3</td>
<td>6</td>
<td>83.17</td>
<td>99.29</td>
<td>96.88</td>
<td>1.42</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>80.08</td>
<td>98.22</td>
<td>96.75</td>
<td>2.63</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>79.86</td>
<td>98.95</td>
<td>95.60</td>
<td>4.79</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>79.30</td>
<td>98.48</td>
<td>94.77</td>
<td>9.37</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>70.14</td>
<td>99.34</td>
<td>94.63</td>
<td>18.42</td>
</tr>
</tbody>
</table>
since their structures are designed to deal with sequence data. Temporal data is also a kind of sequence data so the outstanding performance of RNNs is reasonable. Besides, our proposed method (Bi-LSTM classifier) is slightly better than method using LSTM classifier over $S_e$, $S_p$, $Acc$. It is noticeable that their speeds are nearly same because dimension of features encoded by SDAE is very small.

From Figure 3 to Figure 5, Bi-LSTM and LSTM always show much better than only softmax regression layer. On DS2, Bi-LSTM ($S_e$: 83.57, $S_p$: 99.01, $Acc$: 97.1) get more accurate results than LSTM ($S_e$: 81.00, $S_p$: 98.85, $Acc$: 96.64). On DS3, Bi-LSTM ($S_e$: 89.25, $S_p$: 99.08, $Acc$: 97.79) can detect positive samples better than LSTM ($S_e$: 79.10, $S_p$: 99.49, $Acc$: 96.83) and softmax ($S_e$: 74.6, $S_p$: 99.38, $Acc$: 96.15).

C. EXPERIMENT 2: APPLICATION OF COST-SENSITIVE LOSS FUNCTION

Cost-sensitive loss function is employed in our proposed architecture to solve imbalance problem. This experiment is to prove effectiveness of cost-sensitive loss function, which can be observed by more detailed result of NSFVQ. Table 6 shows comparison between model with cost-sensitive learning and model without cost-sensitive learning. The reason why we don’t take $CTPH$ as assessment criterion is that $CTPH$ is not primary factor in this scenario.

### TABLE 6: Classified results with cost-sensitive loss function on DS1 and DS2 and DS3

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Cost-sensitive Learning</th>
<th>Without Cost-sensitive Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_e$</td>
<td>$S_p$</td>
</tr>
<tr>
<td>DS1</td>
<td>96.32</td>
<td>99.86</td>
</tr>
<tr>
<td>DS2</td>
<td>82.17</td>
<td>99.54</td>
</tr>
<tr>
<td>DS3</td>
<td>91.89</td>
<td>99.22</td>
</tr>
</tbody>
</table>

In Table 6, it is easy to observe cost-sensitive learning increase performance to some extent. It demonstrates that cost-sensitive learning in unbalanced ECG data is practical.

D. EXPERIMENT 3: RESULTS AND COMPARISON WITH OTHER METHODS

On the other hand, to further assess the performance of our proposed methodology, we compare our results with existing methods which get comparatively good results on DS1. DS1 is the most common database used in arrhythmia classification. Table 7 presents a performance summary of these studies.

### TABLE 7: Performance comparison between related methods for arrhythmia classification on DS1

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Techniques</th>
<th>$S_e$%</th>
<th>$S_p$%</th>
<th>$Acc$%</th>
<th>$CTPH$(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hochreiter et. al.</td>
<td>1997</td>
<td>LSTM</td>
<td>96.32</td>
<td>99.4</td>
<td>97.89</td>
<td>22.02</td>
</tr>
<tr>
<td>Graves et. al.</td>
<td>2005</td>
<td>Bi-LSTM</td>
<td>94.53</td>
<td>99.13</td>
<td>98.36</td>
<td>63.49</td>
</tr>
<tr>
<td>Wang et. al.</td>
<td>2013</td>
<td>PCA 50 + PNN 0.3</td>
<td>46.87</td>
<td>93.55</td>
<td>85.72</td>
<td>5.98</td>
</tr>
<tr>
<td>Qin et. al.</td>
<td>2017</td>
<td>Wavelet Transform</td>
<td>81.47</td>
<td>44.40</td>
<td>88.88</td>
<td>6.23</td>
</tr>
<tr>
<td>Acharya et. al.</td>
<td>2017</td>
<td>CNN + SVM</td>
<td>89.41</td>
<td>96.81</td>
<td>95.56</td>
<td>1.55</td>
</tr>
<tr>
<td>Acharya et. al.</td>
<td>2017</td>
<td>CNN</td>
<td>87.01</td>
<td>98.57</td>
<td>96.63</td>
<td>1.55</td>
</tr>
<tr>
<td>Tan et. al.</td>
<td>2018</td>
<td>CNN + LSTM</td>
<td>86.25</td>
<td>99.12</td>
<td>96.96</td>
<td>331.17</td>
</tr>
<tr>
<td>Sannino et. al.</td>
<td>2018</td>
<td>DNN</td>
<td>95.67</td>
<td>99.62</td>
<td>98.95</td>
<td>0.91</td>
</tr>
<tr>
<td>Yildirim et. al.</td>
<td>2018</td>
<td>Wavelet + Bi-LSTM</td>
<td>-</td>
<td>-</td>
<td>99.39</td>
<td></td>
</tr>
<tr>
<td>Hou et. al.</td>
<td>2019</td>
<td>SDAE + AE + LSTM</td>
<td>97.63</td>
<td>99.66</td>
<td>97.95</td>
<td>1.09</td>
</tr>
<tr>
<td>Ours</td>
<td>2019</td>
<td>Bi-LSTM</td>
<td>96.32</td>
<td>99.86</td>
<td>98.25</td>
<td>1.30</td>
</tr>
</tbody>
</table>

It is explicit that proposed method has significant performance: $S_e$:96.32, $S_p$:99.86, $Acc$:98.85 and $CTPH$:1.3. Compared with traditional machining learning methods such as PCA+PNN, deep learning model is simpler and more powerful in all conditions. Deep learning models omit complex extraction of features. Meanwhile, the efficiency of our method is also higher than most of methods. Some methods could get weak advantage in $S_p$ and $CTPH$, but our proposed model gets best performance in $S_p$. However, Yildirim [31] gets the highest accuracy in the summary but its usage of database is different from others. There is still no evidence proving that features extracted manually is better than features extracted by neural networks. This comparison validates that cost-sensitive learning is meaningful for class imbalance problem. Although Bi-LSTM and LSTM are also outstanding, their long computing time limit their application in the reality.

FIG 6 depicts computing time of each method when they compute per heartbeat of DS1. It is not hard to observe that our method has comparatively high accuracy and low computing time(1.3 millisecond), which validates it is practical in reality.

V. CONCLUSION

In this paper, a novel approach for ECG signals classification is proposed by us. We build a cost-sensitive classifier based on SDAE, Bi-LSTM and softmax regression layer to classify ECG signals well. It is an end-to-end model. There are
several advantages: (i) It omits complex extraction of features such as wavelet transformation, and it does not demand researchers professional skills in ECG classification; (ii) SDAE in this approach compresses inputting heartbeats, and thus it saves a lot of running time and boosts computational efficiency; (iii) compared with some traditional classifiers, this classifier containing SDAE is not sensitive to sparse data containing noise, so it is more robust; (iv) to some degree, it relieves class imbalance problem by employing cost-sensitive loss function; (v) in view of the final classified results on three databases, our proposed method has comparatively high accuracy. Generally, SDAE guarantees short time of classification. Bi-LSTM fully utilizes temporal information. Cost-sensitive loss function ensures more accurate results. Overall, for further medical diagnosis in ECG information, Cost-sensitive loss function ensures more accurate results. Overall, for further medical diagnosis in ECG information.

FIGURE 6: Computing time per heartbeat of related classified methods on DS1.

REFERENCES


[29] Jeen-Shing Wang, Wei-Chun Chiang, Yu-Liang Hsu, and Ya-Ting C. Yang. Ecg arrhythmia classification using a probabilistic neural network with a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2936525, IEEE Access
Theory and Methodology in Intelligent Computing.
using morphological and dynamic features of ecg signals. IEEE Transac-
[31] Âºzal yAslÃ¼ÅŸm. A novel wavelet sequence based on deep bidirec-
tional lstm network model for ecg signal classification. Computers in
Biology and Medicine, 96, 03 2018.
[32] YÃijksel Âžubay and GÃijlay Tezel. A new method for classification of
ecg arrhythmias using neural network with adaptive activation function.

***