RESEARCH ARTICLE

Journal of Medical Imaging and Health Informatics
Vol. 9, 1–8, 2019

Classification of Heart Sounds Using Fractional Fourier Transform Based Mel-Frequency Spectral Coefficients and Stacked Autoencoder Deep Neural Network

Zaid Abduh[1,*], Ebrahim Ameen Nehary[1], Manal Abdel Wahed[1], and Yasser M. Kadah[2]

[1]*Systems and Biomedical Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt*
[2]*Electrical and Computer Engineering Department, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Heart sounds contain useful information that can help in early diagnose of heart disease. Therefore, the analysis of such signals has been an active research point for many groups. In this work, we present a new processing and classification system for heart sounds. We introduce a new technique to convert the time series representation of heart sound signal into time-frequency heat map representation based on fractional Fourier transform based mel-frequency spectral coefficients. Such representation is then classified using a stacked sparse autoencoder deep neural network. The proposed system is experimentally verified on the heart sounds database of the PhysioNet/Computing in Cardiology Challenge 2016. The proposed system achieves an accuracy of 0.9550 with 0.8930 sensitivity and specificity 0.9700. The average between sensitivity and specificity (score) is 0.9315. The details of the methodology and its implementation are presented and discussed.

Keywords: Phonocardiogram, Heart Sounds, Computer-Aided Auscultation, Fractional Fourier Transform, Mel-Frequency Spectral Coefficients, Stacked Autoencoder Deep Neural Network.

1. INTRODUCTION

Cardiovascular disease (CVD) remains the leading cause of morbidity and mortality with an estimated 17.7 million people worldwide died from CVD-related conditions in 2015, representing 31% of all global deaths.[1] Heart disease patients can be diagnosed by several techniques with the most sophisticated of these involving medical imaging procedures, which are rather costly and cumbersome and hence of limited availability to most people. On the other hand, the simplest CVD diagnostic technique, which is heart sound auscultation, is an old yet very effective diagnostic tool to check the condition of the heart. Patients are usually examined by means of a stethoscope and may then be referred to a cardiologist if abnormality is detected. Early detection of abnormal heart sounds provides precious time for physicians that is much needed to take corrective actions to treat the cause and prevent cardiovascular system disruption.

The phonocardiogram (PCG) is a graphical representation of heart sound. The PCG signal contains useful information that helps diagnose heart disease and assess the quality of cardiovascular system function.[2] Each PCG is comprised of more than one repeating cardiac cycle (beat), and each beat is comprised of four heart sound states (S1, systole, S2, and diastole) resulting from the closing of the valves at each heart period (mitral and tricuspid valves before systole, aortic and pulmonic valves before diastole). Generally speaking, heart sounds are often difficult to interpret due to their low intensity and dominating frequencies near the lower limits of the human hearing range. Consequently, auscultation needs a lot of training and experience to allow early detection of abnormalities.

A potential solution proposed to address PCG diagnosis problem was to utilize computers to develop automated diagnostic tool to assist the physician in the initial diagnosis or the so-called computer-aided auscultation. In the past few decades, many automated analysis algorithms were developed to assess patients based on the PCG alone without electrocardiogram (ECG) synchronization. There were difficulties associated with such approach that include variation of heart rate in same patients that generate temporal variations of PCG and the limited model generalization across patients. So, this area has received a lot of research work aimed at overcoming such difficulties.

Potes et al.[3] proposed an ensemble of a feature-based classifier and a deep learning based classifier to boost the classification performance of heart sounds. A total of 124 time-frequency features were extracted from PCG signal and input to AdaBoost

*Author to whom correspondence should be addressed.

ensemble classifier. PCG signals were decomposed into four frequency bands to train convolutional neural network (CNN). The final decision was based on combining the outputs of AdaBoost and the CNN. An algorithm developed by Rubin et al.[4] transformed one-dimensional PCG signal into two-dimensional time-frequency heat map representations using mel-frequency cepstral coefficients (MFCC). Convolutional neural network was used to automatically classify normal versus abnormal heart sound recordings.

The PhysioNet/Computing in Cardiology Challenge 2016 offered a dataset to develop, test and compare various algorithms to classify heart sounds. Many research groups contributed new methods in response to this challenge.[5–7] Gokhale[8] presented an algorithm that uses Hilbert envelope and wavelet features. The boosted trees ensemble classifier using LogitBoost was applied for automated PCG signals classification. Goda et al.[9] used support vector machine (SVM) classifier and time-frequency domain features. An algorithm was developed by Grzegorczyk et al.[10] to determine normal-abnormal heart sound recording based on neural networks. Forty-eight time and frequency domain features were used with conventional neural network and auto-encoder deep neural network. Another technique was presented by Tschannen et al.[11] for heart sounds classification where deep features generated by a wavelet-based convolutional neural network and time-frequency domain features were used with L_2-SVM classifier. Homsi et al.[12] introduced an approach for classifying PCG signals using time, frequency, wavelet and statistical domain features with nested set ensemble classifiers that included random forest, LogitBoost and cost-sensitive classifiers. Langley and Murray[13] classified unsegmented and short duration PCG signals using wavelet entropy where a wavelet entropy threshold was determined from the training set then PCG signals with entropy below the threshold were classified as abnormal. In Singh-Miller and Singh-Miller,[14] spectral features with discriminative model based on random forest regressor were applied for classification of heart sound recordings. Vernekar et al.[15] implemented Markov features with a weighted ensemble classifier including four AdaBoost ensemble classifiers and four artificial neural networks (ANN) to classify PCG signals. Plesinger et al.[16] demonstrated a fuzzy logic like approach with logical rules and probability assessment based on histograms to classify heart sounds. A frequency-domain bandpass filter and Hilbert transform were used to derive amplitude envelope in five frequency bands. The averaged shapes of S1/S2 pair were computed from amplitude envelopes, then a total of 228 features extracted from statistical and symmetry properties of the averaged shapes. Nabhan and Warriek[17] tried to improve the work in Ref. [12] where the outlier signal was detected and separated from standard range signal by using an interquartile range threshold. A total of 131 features were extracted from the standard and outlier signals were fed separately into an ensemble of 20 two-step classifiers. In the first step, the classifier included a nested set of ensemble algorithms, consisting of a cost-sensitive classifier (CSC), LogitBoost (LB) and random forest (RF) classifiers. The second step used a voting rule for the class labels from the first step. Abdollahpur et al.[18] proposed an algorithm that assessed the signal quality of the segmented cardiac cycle then a total of ninety features including time domain, time-frequency, perceptual and mel-frequency cepstral coefficient (MFCC) were extracted from the correctly segmented cycles only. The classification was performed using three feed-forward neural networks followed by a voting system. Langley and Murray[19] tried to improve the work in Ref. [13] where short and unsegmented heart sounds recordings were classified using feature threshold-based classifier. Spectral amplitude and wavelet entropy features were calculated using FFT and wavelet analysis. For each feature, a threshold-based classifier was built by analysis of frequency and scale to determine the optimal threshold for classification accuracy. A decision tree was then used to combine the spectral amplitude and wavelet entropy. Maknickas and Maknickas[20] applied CNN to classify heart sound records with mel-frequency spectral coefficients (MFSC), difference and second-order difference of the MFSC calculated and fed to CNN as three dimensions for each frame. Whitaker et al.[21] proposed combining sparse coding and time domain features for heart sounds classification. Springer's segmentation was used to separate each record into five arrays of smaller audio segments. The first four arrays contained a list of all S1, systole, S2 and diastole sounds respectively. The fifth array contained copies of the full heart cycles. A discrete Fourier transform was calculated for each sound segment and sparse coding was applied whereby frequency-domain data are decomposed into a dictionary matrix and a sparse coefficient matrix. Five SVM classifiers were trained for each audio segment as well as the full cardiac cycle and a sixth SVM combined the preliminary SVMs.

In spite of the success of previous methods to reach significant improvement in the accuracy of classification of heart sounds, there is still room for further improvement of results and robustness of automated diagnostic systems. In this work, a new computer-aided auscultation system is presented. The new system utilizes novel features obtained from the fractional Fourier transform based Mel-frequency spectral coefficients. The fractional Fourier transform offers a generalization of the Fourier transform with the Fourier spectrum and the time domain signal are both special cases of this transform. Hence, it allows a more flexible time-frequency representation than other methods such as the spectrogram, Wigner distribution or ambiguity function in that it can transform signals to any intermediate domain between time and frequency.[22] Such time-frequency representation of heart sound signals are then utilized as input to stacked autoencoder deep neural network (DNN). The implementation of the proposed diagnostic system is described in terms of its preprocessing, heart sound segmentation, transformation of time series waveforms into time frequency heat map representations and classification stages. The proposed system with several variants of its implementation are verified on the dataset of the PhysioNet/Computing in Cardiology Challenge 2016 and compared to previous work on the same dataset.

2. METHODOLOGY

The components of the proposed computer aided auscultation system are shown in Figure 1 and are detailed as follows.

2.1. Preprocessing

The heart sounds comprise several key components called S1, S2, S3 and S4 which overlap with each other in the frequency domain. Moreover, murmurs and artifacts from respiration and other non-physiological events also overlap significantly within the same frequency range. The typical frequency ranges of these components are: S1 within 10–140 Hz (energy concentration usually in low frequencies of 25–45 Hz), S2 within 10–200 Hz

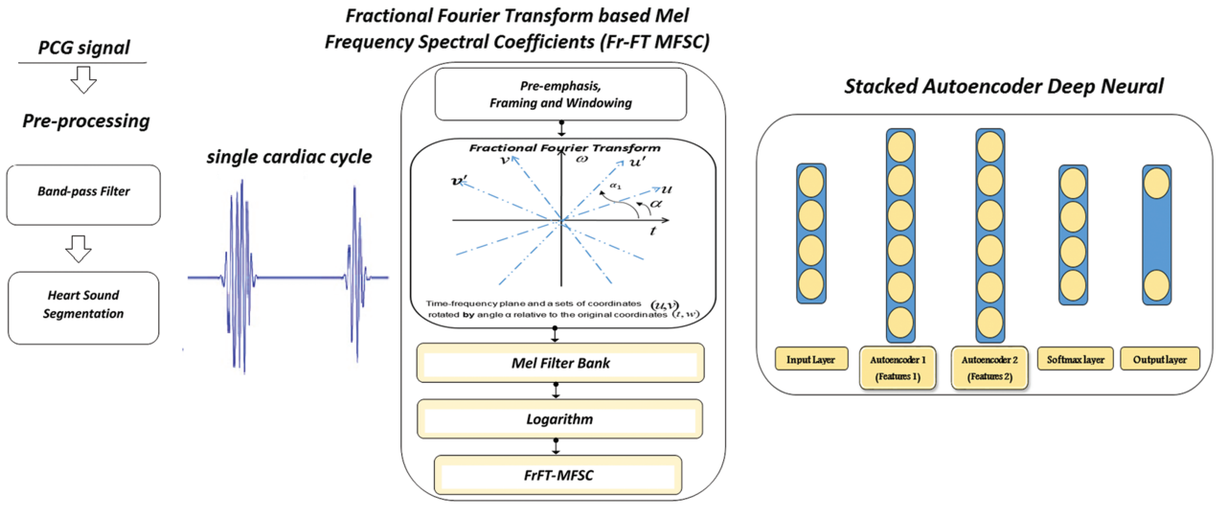

Fig. 1. Block diagram of the proposed approach for classification of heart sounds using stacked autoencodder deep neural network.

(energy concentration usually in low frequencies of 55–75), S3 and S4 within 20–70 Hz, murmurs can be as high as 600 Hz, respiration for 200–700 Hz. The fundamental heart sound components that need to be segmented (mainly S1 and S2) overlap with many noise sources in the frequency domain, which leads to difficulty in separation of heart sounds from abnormal sounds or artifacts using traditional frequency-domain analysis. Moreover, the morphological similarity of the noise to the fundamental heart sounds makes identification of the latter also extremely difficult using time-domain techniques. It should also be noted that abnormal heart sounds usually show higher frequencies, with noise between beats. On the other hand, normal heart sounds are more regular, with silence between beats.[7]

Ambient sounds, lung sound, internal body noise, cough and stethoscope movement are the main interferences in heart sounds recording and analysis. Therefore, the first step should use an effective filtration method to enhance the heart sounds signal by reducing the influence of background noise and removing spike noise. In this work, a 3rd-order Butterworth band pass filter with corner frequencies of 15 and 800 Hz is used to select the useful bandwidth of the heart sounds that includes the energy concentration ranges of its components in addition to the range of frequencies of murmurs.

2.2. Heart Sounds Segmentation

Accurate segmentation of heart sound is an important step to extract useful features for better classification. In the second stage, each PCG signal is split into the fundamental heart sounds (S1, Systole, S2 and Diastole) using Springer's improved version of Schmidt's segmentation algorithm.[23] Then, each complete cardiac cycle is used for processing. This algorithm uses a logistic regression hidden semi-Markov model (HSMM) to predict the most likely sequence of states by incorporating information about expected heart sound state durations.

To mitigate the problem of variable time length of cardiac cycles (and hence size of their digital signals) in subsequent processing steps, the size of all signals was set to be the longest cardiac cycle found across all PCG recordings (here, it was around 2 s). For cardiac cycles with shorter length, they were zero-padded to that length. Since the sampling period remains the same, this amounts to a higher resolution sampling of the unit circle in the z-domain. This ensures the alignment of the spectral components computed from all samples of varying record lengths given the uniform frequency resolution.[24]

2.3. Fractional Fourier Transform Based Mel-Frequency Spectral (FrFT-MFSC)

A modified version of MFCC will be implemented to convert the time series representation of the segmented heart sound signal into time-frequency heat map representations. This will be done using the Fractional Fourier Transform (FrFT), which is a generalization of Fourier transform and indicates a rotation of a signal in time-frequency plane. Whereas the Fourier transform can obtain the frequency components of a signal, FrFT analysis can show the mixed time and frequency components of the signal. Therefore, FrFT is suitable for non-stationary signal processing and has wide applications in basic signal analysis and speech recognition.

The continuous form of FrFT with ath-order of a signal $s(t)$ can be defined within $0 \leq |a| \leq 2$ through the linear operator as,[22, 26, 28–29]

$$(F^a s)(w_a) = \int_{-\infty}^{\infty} K_a(w_a, t) s(t)\, dt \tag{1}$$

Here, the kernel $K_a(w_a, t)$ is given by:

$$K_a(w_a, t) = \begin{cases} k_{\varnothing} \exp(j\pi(w_a^2 \cot(\varnothing) \\ \quad -2w_a t \csc(\varnothing) + t^2 \cot(\varnothing))), & \text{if } a \neq 0, \pm 2, \\ \delta(w_a - t), & \text{if } a = 0, \\ \delta(w_a + t), & \text{if } a = \pm 2 \end{cases} \tag{2}$$

where, $\varnothing = a\pi/2$ and $k_{\varnothing} = \exp(-j((\pi sgn(\varnothing)/4) - (\varnothing/2))/\sqrt{|\sin(\varnothing)|} = \sqrt{1 - j\cot(\varnothing)}$, and w_a means the variables in ath-order fractional Fourier transform. Similar to the discrete Fourier transform (DFT), the discrete fractional Fourier

transform (DFrFT) matrix $F^a(m,n)$ is obtained as the discrete version of Eq. (2) as,

$$F^a(m,n)=\sum_{k=0}^{N-1}u_k(m)\exp\left(\frac{-j\pi ka}{2}\right)u_k(n) \tag{3}$$

Here, u is discrete Hermite-Gaussian function and a is the fractional order. The discrete fractional Fourier transform of a signal is just the matrix vector multiplication of this transform matrix in Eq. (3) with the signal vector.

The fractional Fourier transform based Mel-frequency spectral coefficients (FrFT-MFSC) features transform the original PCG signal into a time-frequency representation of the distribution of signal energy same as the traditional Mel-frequency cepstral coefficients (MFCC). The discrete fractional Fourier transform matrix in Eq. (3) will be used instead of discrete Fourier transform matrix and the log-energy will be computed directly from the Mel-frequency spectral coefficients without using the discrete cosine transform.

FrFT-MFSC features are computed by the following steps:
1. The signal is pre-emphasized then framed and windowed into 125 ms frames length.
2. The discrete fractional Fourier transform of the windowed signal is calculated.
3. The powers of the obtained spectrum are mapped into the mel-scale using triangular overlapping windows.
4. The logarithms of the powers are calculated at each mel-frequency to obtain the FrFT-MFSC features.

It should be noted that normalization is performed to make all FrFT-MFSC coefficients in the range of [0, 1]. The Mel-scale is defined at a given frequency f in Hz as,

$$Mel(f)=2595\log_{10}\left(1+\frac{f}{700}\right) \tag{4}$$

The Mel-frequency scale is approximately linear for frequencies below 1 kHz and logarithmic for frequencies above 1 kHz.[25] This is motivated by the fact that the human auditory system becomes less frequency-selective as frequency increases above 1 kHz.

2.4. Stacked Sparse Autoencoders Deep Neural Network

An Autoencoder (AE), also named autoassociator or Diabolo neural network, derived from the multi-layer perceptron (MLP), composed by an input layer, a hidden layer and an output layer.[30] The autoencoder aims to transform inputs into outputs with the least possible amount of distortion.[31] It maps an input $x\in[0,1]^d$ to a hidden representation $y\in[0,1]^{d'}$ through a deterministic mapping (encoding),

$$y=s(Wx+b) \tag{5}$$

where s is a transfer function for the encoder and it is non-linear function such as a sigmoid function, e.g.: $s(t)=1/(1+e^{-t})$, or a positive saturating linear function

$$s(t)=\begin{cases}0, & \text{if } t\le 0,\\ t, & \text{if } 0<t<1,\\ 1, & \text{if } t\ge 1\end{cases}$$

and where W is a weight matrix and b is a bias vector for encoder. The hidden representation y, or code, is then mapped back into a reconstruction z of the same shape as x (decoding). The mapping happens through linear transformation or a positive saturating linear function or non-linear transformation such as a sigmoid function,

$$z=s'(\bar{W}y+\bar{b}) \tag{6}$$

where z is a prediction of x, given the code y. where s' is a transfer function for the decoder, $\bar{W}$ is a weight matrix and $\bar{b}$ is a bias vector for decoder. In general, z is not to be interpreted as an exact reconstruction of x, but rather in probabilistic terms as the parameters (typically the mean) of a distribution $p(X\mid Z=z)$ that may generate x with high probability. Training an autoencoder is unsupervised since that no labeled data is needed. The training process is still based on the optimization of a cost function. Optimization process for the model parameters $(W,\bar{W},b,\bar{b})$ is done to minimize the average reconstruction error. The cost function that measures the squared error in most traditional autoencoders is given by,

$$L(x,z)=\|x-z\|^2 \tag{7}$$

The basic autoencoders is forced to learn the identity function. If the hidden layer size is greater than the size of the input layer, a sparsity constraint imposed to get more robust features.[30–33] Sparsity of an autoencoder is possible by adding a regularizer to the cost function in Eq. (7). This regularizer is a function of the average output activation measure of a neuron. The average output activation measure of a neuron i is defined as,

$$\hat{\rho}_i=\frac{1}{n}\sum_{j=1}^{n}z_i^{(1)}(x_j)=\frac{1}{n}\sum_{j=1}^{n}h(w_i^{(1)T}x_j+b_i^{(1)}) \tag{8}$$

where $\hat{\rho}_i$ is the average output activation measure, n is the total number of training examples, x_j is the jth training example, $w_i^{(1)T}$ is the ith row of the weight matrix $W^{(1)}$, and $b_i^{(1)}$ is the ith entry of the bias vector, $b^{(1)}$.

Two terms are added, namely L_2 regularization term and the sparsity regularization term. So, the cost function for training a sparse autoencoder is defined as,

$$L(x,z)=\overbrace{\|x-z\|^2}^{\text{mean squared error}}+\lambda*\overbrace{\Omega_{\text{weights}}}^{L_2\text{ regularization}}+\beta*\overbrace{\Omega_{\text{sparsity}}}^{\text{sparsity regularization}} \tag{9}$$

where λ is the coefficient for L_2 regularization term and β is the coefficient for the sparsity regularization term. The L_2 regularization term is defined as,

$$\Omega_{\text{weights}}=\frac{1}{2}\sum_{l}^{L}\sum_{j}^{n}\sum_{i}^{k}(w_{ji}^{l})^2 \tag{10}$$

where L is the number of hidden layers, n is the number of training examples, and k is the number of variables in the training data. The sparsity regularization term could be calculated using the Kullback-Leibler divergence as,

$$\Omega_{\text{sparsity}}=\sum_{i=1}^{D(1)}KL(\rho\|\hat{\rho}_i)=\sum_{i=1}^{D(1)}\rho\log\left(\frac{\rho}{\hat{\rho}_i}\right)+(1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_i}\right) \tag{11}$$

where ρ is the desired value of the average output activation measure also defined as sparsity proportion and $\hat{\rho}_i$ the measured

Table I. Parameters of the stacked sparse deep neural network.

Parameter	Value
Autoencoder	2
Training softmax layer	Mean square error
Hidden size	800
L_2 weight regularization	0.00001
Sparsity regularization	4
Sparsity proportion	0.02
Encoder transfer function	Positive saturating linear
Decoder transfer function	Linear

value of the average output activation measure for neuron i. In training we would like to make the value of $\hat{\rho}_i$ for hidden neuron i close to the value of ρ. The coefficient for L_2 regularization term λ, the coefficient for the sparsity regularization term β and the sparsity proportion ρ, are the set parameters for autoencoder training.

Autoencoders can be stacked in a greedy layer-wise fashion to form a deep neural network that is the stacked sparse autoencoders deep neural network where each level is associated with an autoencoder that can be trained separately. The output of the autoencoder at the first layer feeds as input to the autoencoder in the second layer and so on. When all layers are pre-trained, a classification layer is added, and the deep network can be fine-tuned. Each stacked sparse autoencoders deep neural network has input layer, autoencoders layers followed by a softmax layer and the output layer. We can use more than one autoencoder. Greedy layer-wise training approach used to obtain good parameters.[27]

In this work, we build a deep neural network classifier using two sparse autoencoder layers and a softmax layer. In training phase, the autoencoder in the first layer trains on raw input data to obtain the set of parameters $(W_1, \bar{W}_1, b_1, \bar{b}_1)$, then extract the primary features in the hidden layer based on raw input. The primary features from first autoencoder used as an input to train the second autoencoder and obtain the second set of parameters $(W_2, \bar{W}_2, b_2, \bar{b}_2)$, then extract the secondary features in the hidden layer of the second autoencoder based on these primary features. The secondary features from the second autoencoder used to train a softmax layer for classification. After that stack the two autoencoders and the soft max layer to form a deep neural network, we finally train the deep network using the raw input data. Then, the deep network will be ready to predict the output for any input. Table I summarizes the set of parameters considered in this work.

2.5. Performance Evaluation

To evaluate the performance of classification process, the confusion matrix is computed and used to calculate the values for sensitivity, specificity and accuracy as,

$$\text{Sensitivity (Se)} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{12}$$

$$\text{Specificity (Sp)} = \frac{\text{TN}}{\text{TN}+\text{FP}} \tag{13}$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{14}$$

Here TP, TN, FP, and FN are the confusion matrix entries representing true positive, true negative, false positive and false negative cases respectively. An additional performance metric called the score conventionally used in the area of heart sounds classification. It is defined as the average of the sensitivity and specificity metrics.

3. EXPERIMENTAL VERIFICATION

The performance of the proposed system was verified using the data of the PhysioNet/Computing in Cardiology Challenge 2016 database.[7] This dataset includes 3153 recordings. The sure labeled data of the training dataset includes 2868 recordings collected from six research groups. Normal patient records are 2249 whereas abnormal patient records are 619, lasting from 5 seconds to just over 120 seconds. All recordings have been resampled to 2,000 Hz and have been provided as ".wav" format. They were recorded in different real-world clinical and nonclinical environments and include recordings of varying amounts of noise. They were collected from healthy people and from patients who suffered from a variety of illnesses, including heart valve defects (mitral valve prolapse, mitral regurgitation, aortic stenosis, valvular surgery) and coronary artery disease. The data recorded from different locations on the body (including aortic area, pulmonic area, tricuspid area and mitral area, among others). The data are clearly imbalanced since the number of normal recordings are much larger than that of abnormal recordings. Since the testing dataset of the challenge was not available from the challenge organizers, the available training dataset was divided randomly into two independent sets with 80% for training and 20% for testing. This assumes that the missing testing dataset was taken from the same population as the training data and thus expected to possess the same characteristics and challenges such as being imbalanced.

Each of the PCG records was preprocessed using the Butterworth band-pass filter of order 3 with corner frequencies 15 and 800 Hz. Each record was segmented into cardiac cycles resulting in 79492 cardiac cycles. The longest cardiac cycle found across all PCG recordings has a length of around 2 s. So, if a cardiac cycle had a length less than 2 seconds, the time series was zero-padded to that length.

For FrFT-MFSC time frequency heat map transformation, the processing parameters were as follows: four fractional orders $(a = 0.90, 0.95, 1.0, 1.1)$. The frame length parameter was chosen to cover the duration of each of the fundamental heart sounds S1 and S2 (nominal values: S1 = 122 ms, S2 = 114 ms). So, the chosen frame length is taken as 125 ms. The frame shift was taken close to 50% of the frame length as is commonly used in the literature and chosen to be 50 ms. The number of bands was taken as 20 based on experimentation. For each fractional order, a 20×38 spectral coefficients matrix was computed. The time frequency heat map representations vector was formed by concatenating spectral coefficients matrices for all four fractional orders. That is, for each cardiac cycle, a total of 3040 features were obtained. To improve the classification process, all features vector values were normalized to interval [0, 1]. Then, dimension reduction process was performed using PCA such that the percentage of variance represented by the results is 95%. This process reduced the size of vector down to 40.

At the beginning, sparse autoencoders was trained on the training data without using labels. Greedy layer-wise training approach used so the first autoencoder was trained using the training set with a hidden layer of size 800, a positive saturating

linear transfer function for the encoder and a linear transfer function for the decoder. Then the features in the hidden layer were extracted. After training the first autoencoder the second autoencoder was trained in a similar way by using the obtained features from the first autoencoder. The features in the hidden layer of the second autoencoder was extracted. The second autoencoder have the same parameter as first one. Then a softmax layer was trained to classify the obtained features vector from the second autoencoder. Unlike the autoencoders, training of the softmax layer is done in a supervised approach utilizing the labels for the training data. A standard softmax cross entropy loss function was used to optimize the network during training. Finally stack the two encoders and the softmax layer to form the deep network. The deep network is ready for more analysis using the test data.

The performance of sparse autoencoder is controlled by adjusting the L2 weight regularizer coefficient, sparsity regularizer coefficient and sparsity proportion factor. In this work, different values for L2 weight regularizer coefficient were checked but no noticeable change in the performance of autoencoders so L2 weight regularizer coefficient was fixed to 0.00001. Performance comparison was done by trying different values for sparsity regularizer coefficient and sparsity proportion factor.

4. RESULTS AND DISCUSSION

Table II shows the results of using different values of sparsity regularizer coefficient and sparsity proportion factor was set to 0.02. Table III shows the results of using different values of sparsity proportion factor and sparsity proportion factor was fixed to 4. Sparsity regularizer coefficient of 4 and sparsity proportion factor of 0.02 achieve the best result, accuracy of 0.9550 with 0.8930 sensitivity and specificity 0.9700. The average between sensitivity and specificity (score) is 0.9315.

In order to establish the generalization capability of the proposed method, multiple random sampling trials for local hold-out testing (here 80%–20%) are performed to verify that the proposed algorithm generates robust results and not by chance. The results of five such experiments are shown in Table IV with the accuracy, sensitivity, specificity and score computed for each experiment. Also, the mean and standard deviation of the results from all five experiments are calculated. As can be observed, the results vary within a very narrow range as indicated by the standard deviation of all experiments, which were 0.14% for the accuracy, 0.5% for the sensitivity, 0.1% for the specificity, and 0.24% for the score relative to their respective mean values. This rules out the possibility that the reported results were affected by chance and confirms the robustness of the methodology.

The comparison of the results of the proposed work against those from previous work in the literature is presented in Table V.

Table II. Results of using different values of sparsity regularizer coefficient at sparsity proportion factor of 0.02.

Stacked AE sparsity regularization	Accuracy	Sensitivity	Specificity	Score
2	0.9510	0.8740	0.9700	0.9220
4	0.9550	0.8930	0.9700	0.9315
6	0.9520	0.8810	0.9700	0.9255
8	0.9530	0.8840	0.9690	0.9265
10	0.9530	0.8870	0.9690	0.9280
16	0.9530	0.8830	0.9700	0.9265

Table III. Results of using different values of sparsity proportion factor at sparsity regularizer coefficient of 4.

Stacked AE sparsity proportion	Accuracy	Sensitivity	Specificity	Score
0.05	0.9446	0.8610	0.9650	0.9130
0.04	0.9465	0.8720	0.9650	0.9185
0.03	0.9506	0.8760	0.9690	0.9225
0.02	0.9550	0.8930	0.9700	0.9315
0.01	0.9558	0.8890	0.9725	0.9308

The table shows the proposed method as they rank against other techniques as sorted by the average between sensitivity and specificity (score). It also shows the classification methods used in each and the sensitivity and specificity values when available. The comparison of such methods should address the balance between sensitivity and specificity values. As can be observed, the proposed method shows better performance than the other techniques included in this comparison. Also, the sensitivity and specificity values obtained from the new system are relatively close. This indicates the potential of the new approach for use in clinical applications.

This study indicates the value of FrFT-MFSC features for phonocardiogram analysis where FrFT-MFSC represent the distribution of PCG signal energy in a more effective time-frequency manner even under different noise and recording environment conditions. Applying FrFT-MFSC features with different fractional orders provides a good tool to represent noisy PCG signal in different time-frequency planes where they also preserve locality in both time and frequency. The use of log-energy computed directly from the mel-frequency spectral coefficients maintains such time-frequency localization. The alternative yet more common use of discrete cosine transform for this computation projects the spectral energies into a new global basis that would not maintain such localization.

In conducting this study, several difficulties were encountered. First, the lack of large enough dataset for deep neural network training to get more robust results was a major problem. This was overcome by breaking down the heart sound signals into segments representing cardiac cycles to significantly increases the number of training data. Second, variations in heart rate lead to temporal record length variations for cardiac cycle segments. This was addressed via zero-padding to the length of the longest record. This maintains the same sampling rate while harmonizing frequency resolution among all records, which is critical to our method given its reliance on frequency domain features.[24] Finally, the variability between records that is introduced by heterogeneity in the collection of the recordings can render a classifier trained on one population much less effective when applied

Table IV. Results of multiple random sampling trials for local hold out testing showing.

Run index	Accuracy	Sensitivity	Specificity	Score
1	0.9531	0.881	0.9720	0.9265
2	0.9530	0.8860	0.9700	0.9280
3	0.9530	0.8840	0.9700	0.9270
4	0.9550	0.8930	0.9700	0.9315
5	0.9560	0.8910	0.9720	0.9315
Mean	0.9540	0.8870	0.9708	0.9289
Standard deviation	0.0014	0.0049	0.0011	0.0024

Table V. Comparison of proposed method to those reported in literature for the same dataset.

Method	Classifier	Reported results		
		Sensitivity	Specificity	Score
Proposed work	FrFT-MFSC with stacked autoencoder deep neural network	0.8930	0.9700	0.9315
Plesinger et al.[16]	Fuzzy logic like approach	0.8690	0.9370	0.9030
Langley and Murray[19]	Amplitude spectrum and wavelet entropy threshold followed by decision tree	0.8690	0.9370	0.9030
Goda and Hajas[9]	SVM	0.9700	0.8200	0.9000
Nabhan and Warriek[17]	Ensemble methods with outliers for phonocardiogram classification	0.8740	0.9140	0.8940
Abdollahpur et al.[18]	Three feed-forward NNs	0.8883	0.8851	0.8867
Whitaker et al.[21]	SVM	0.8867	0.8816	0.8841
Homsi et al.[12]	Nested ensemble of algorithms including random forest, LogitBoost and a cost-sensitive classifier	0.9440	0.8690	0.8840
Tschannen et al.[11]	L2-SVM	0.9080	0.8320	0.8700
Potes et al.[3]	Final decision rule based on AdaBoost ensemble classifier and convolutional neural network	0.8800	0.8200	0.8500
Singh-Miller et al.[14]	Discriminative model based on random forest regressor	0.8100	0.8900	0.8500
Potes et al.[3]	Convolutional neural network (CNN)	0.7900	0.8600	0.8200
Vernekar et al.[15]	Weighted ensemble classifier including four AdaBoost ensemble and four ANN classifiers	0.7920	0.8430	0.8200
Potes et al.[3]	AdaBoost-ensemble classifier	0.7000	0.8800	0.7900
Langley and Murray[13]	Wavelet entropy threshold	0. 9500	0.6000	0.7800
Gokhale[8]	Boosted trees ensemble classifier	NA*	NA*	0.7600
Grzegorczyk et al.[10]	Conventional neural network and autoencoder	0.8300	0.6200	0.7300

Note: *NA: Not available from the reference.

to another. Here, the features proposed seem to capture the salient features among all such populations and hence show a more robust performance among populations coming from very different collections settings.

The processing platform used in this work was based on an Intel® Core™-i7 laptop with 8 GB of RAM running Matlab 2017a on Windows 10 operating system. The processing time for the training part of the study was around 3.5 hours, while that of the testing phase was around 1.8 seconds. It should be noted that the testing phase processing time indicates the potential of the system for real-time performance as it stands. Several approaches can make such processing time much smaller including the use of multi-core processing, use of graphical processing unit processing, and/or implementation of code in C++.

It should be noted that the main focus of this work was to show the potential of the new method and hence the optimization of performance under different sets of algorithm parameters such as the set of fractional orders, frame length, frame shift and number of warped bands was not thoroughly pursued. This remains to be addressed in future work.

5. CONCLUSIONS

A new approach is proposed to classify heart sounds. The main contribution of this approach is to introduce new features based on fractional Fourier transform based Mel-frequency spectral coefficients combined with stacked autoencoder deep neural network. The description of the proposed methodology and its implementation details are presented. The results of experimental verification indicate that the presented approach has potential to overcome the challenges encountered during heart sound classification under different settings. These results indicate the robustness of the new features and their potential in clinical use.

References and Notes

1. WHO, Cardiovascular diseases world statistics on WHO, Last updated May 2017, accessed October **(2017)**, www.who.int/mediacentre/factsheets/fs317/en/.
2. C. Jian, G. Xingming, and X. Shouzhong, Study on the signification and method of heart sound recognition. *Foreign Medical Biomedical Engineering Fascicle* 27, 87 **(2004)**.
3. C. Potes, S. Parvaneh, A. Rahman, and B. Conroy, Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 621–624.
4. J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, and K. Sricharan, Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 813–816.
5. G. D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, et al., Classification of normal/abnormal heart sound recordings: The physionet/computing in cardiology challenge 2016, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 609–612.
6. G. D. Clifford, C. Liu, B. Moody, J. Millet, S. Schmidt, Q. Li, et al., Recent advances in heart sound analysis. *Physiological Measurement* 38, E10 **(2017)**.
7. C. Liu, D. Springer, Q. Li, B. Moody, R. A. Juan, F. J. Chorro, et al., An open access database for the evaluation of heart sound algorithms. *Physiological Measurement* 37, 2181 **(2016)**.
8. T. Gokhale, Machine learning based identification of pathological heart sounds, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 553–556.
9. M. A. Goda and P. Hajas, Morphological determination of pathological PCG signals by time and frequency domain analysis, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 1133–1136.
10. I. Grzegorczyk, M. Soliński, M. Łepek, A. Perka, J. Rosiński, J. Rymko, et al., PCG classification using a neural network approach, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 1129–1132.
11. M. Tschannen, T. Kramer, G. Marti, M. Heinzmann, and T. Wiatowski, Heart sound classification using deep structured features, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 565–568.
12. M. N. Homsi, N. Medina, M. Hernandez, N. Quintero, G. Perpiñan, A. Quintana, et al., Automatic heart sound recording classification using a nested set of ensemble algorithms, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 817–820.
13. P. Langley and A. Murray, Abnormal heart sounds detected from short duration unsegmented phonocardiograms by wavelet entropy, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 545–548.

14. N. E. Singh-Miller and N. Singh-Miller, Using spectral acoustic features to identify abnormal heart sounds, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 557–560.
15. S. Vernekar, S. Nair, D. Vijaysenan, and R. Ranjan, A novel approach for classification of normal/abnormal phonocardiogram recordings using temporal signal analysis and machine learning, *Proc. 2016 Computing in Cardiology Conference (CinC)* **(2016)**, pp. 1141–1144.
16. F. Plesinger, I. Viscor, J. Halamek, J. Jurco, and P. Jurak, Heart sounds analysis using probability assessment. *Physiological Measurement* 38, 1685 **(2017)**.
17. H. M. Nabhan and P. Warrick, Ensemble methods with outliers for phonocardiogram classification. *Physiological Measurement* 38, 1631 **(2017)**.
18. M. Abdollahpur, A. Ghaffari, S. Ghiasi, and M. J. Mollakazemi, Detection of pathological heart sounds. *Physiological Measurement* 38, 1616 **(2017)**.
19. P. Langley and A. Murray, Heart sound classification from unsegmented phonocardiograms. *Physiological Measurement* 38, 1658 **(2017)**.
20. V. Maknickas and A. Maknickas, Recognition of normal-abnormal phonocardiographic signals using deep convolutional neural networks and mel-frequency spectral coefficients. *Physiological Measurement* 38, 1671 **(2017)**.
21. B. M. Whitaker, P. B. Suresha, C. Liu, G. Clifford, and D. Anderson, Combining sparse coding and time-domain features for heart sound classification. *Physiological Measurement* 38, 1701 **(2017)**.
22. L. B. Almeida, The fractional Fourier transform and time-frequency representations. *IEEE Transactions on Signal Processing* 42, 3048 **(1994)**.
23. D. B. Springer, L. Tarassenko, and G. D. Clifford, Logistic regression-HSMM-based heart sound segmentation. *IEEE Transactions on Biomedical Engineering* 63, 822 **(2016)**.
24. L. Chapparo, Signals and Systems Using Matlab, 2nd edn., Academic Press **(2015)**.
25. D. O'Shaughnessy, Speech Communications: Human and Machine, 2nd edn., Wiley-IEEE Press **(1999)**.
26. A. C. McBride and F. H. Kerr, On namias's fractional fourier transforms. *IMA Journal of Applied Mathematics* 39, 159 **(1987)**.
27. A. Ng, Deep Learning and Unsupervised Feature Learning, http://deeplearning.stanford.edu/wiki/index.php/Main_Page, accessed January **(2018)**.
28. C. Candan, M. A. Kutay, and H. M. Ozaktas, The discrete fractional fourier transform. *IEEE Transactions on Signal Processing* 48, 1329 **(2000)**.
29. S.-C. Pei, M.-H. Yeh, and C.-C. Tseng, Discrete fractional Fourier transform based on orthogonal projections. *IEEE Transactions on Signal Processing* 47, 1335 **(1999)**.
30. Y. Bengio, Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2, 1 **(2009)**.
31. A. Ng, Sparse Autoencoder, CS294A Lecture Notes **(2011)**, Vol. 72.2011, pp. 1–19.
32. C. Poultney, S. Chopra, and Y. L. Cun, Efficient learning of sparse representations with an energy-based model. *Advances in Neural Information Processing Systems* 19, 1137 **(2006)**.
33. H. Lee, C. Ekanadham, and A. Y. Ng, Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems* 20, 873 **(2007)**.

Received: xx Xxxx xxxx. Accepted: xx Xxxx xxxx.