Journal of King Saud University – Computer and Information Sciences 32 (2020) 1149–1157

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# The effect of dictionary learning on weight update of AdaBoost and ECG classification

Mücahid Barstuğan *, Rahime Ceylan

*Konya Technical University, Faculty of Engineering and Natural Sciences, Electrical & Electronics Engineering Department, Konya, Turkey*

ARTICLE INFO

*Article history:*
Received 14 July 2018
Revised 26 October 2018
Accepted 12 November 2018
Available online 14 November 2018

*Keywords:*
AdaBoost
Dictionary learning
ECG
Feature subsets
Signal classification
Sparse representation

ABSTRACT

A signal can be represented by sparse representation with fewer coefficients. Due to this ability, sparse representation is used in research fields such as signal compression, noise elimination, and classification. In this study, sparse coefficients of the signals were obtained by using dictionary learning and sparse representation algorithms. The obtained coefficients were used in the weight update process of three different classifiers, which were created by using AdaBoost, SVM, and LDA algorithms. So, Dictionary learning based AdaBoost classifiers were obtained. The proposed Dictionary Learning (DL) based AdaBoost classifiers classified the ECG (Electrocardiography) signals. Before classification, the feature selection process was applied to ECG signals and six different feature subsets were obtained by Discrete Wavelet Transform (DWT), First Order Statistics (FOS), T-test, Bhattacharyya, Entropy, and Wilcoxon test methods. The feature subsets were used as the new dataset. The classification process was done by the proposed method and satisfying results were obtained. The best classification accuracy was obtained as 99.75% by the proposed dictionary learning based method called as DL-AdaBoost-SVM on feature subsets obtained by DWT and Wilcoxon test methods.



## 1. Introduction

One of the well-known artificial intelligence methods in biomedical data classification is ensemble classifiers. In the ensemble classifiers method, several weak classifiers, which have a lower classification accuracy, are used. Each of classifiers has a weight, which affects to classification result. Obtaining the classification result by using these weights is called "weight voting". AdaBoost is one of the most used ensemble classifier methods, which uses weight-voting process. It was proposed by Freund and Schapire (1997). The AdaBoost algorithm is easy to implement and can increase the classification accuracy. Also, it can be used with different classifiers. Wang and Pineau (2016) combined the AdaBoost and Dictionary Learning algorithms in their study. The sparse coefficients were obtained in the study, and then the weak classifier weights of AdaBoost were updated by these coefficients. It is shown that their method has high performance in classification. Zhang and Zhou (2011) obtained sparse representation coefficients by LP-AdaBoost ensemble classifier. In their study, the *k*-NN algorithm was used as a weak classifier, and sparse representation coefficients were defined as the weights of base learners. Imbalanced dataset was used to observe the classification performance of the method. Huang and Zhang (2014) proposed semi-supervised sparse multi-linear analysis (SSSMDA) method. In the proposed method, a third-degree short time Fourier transform was applied to ECG dataset. During the experiments, it was proven that the method is effective on sparse ECG dataset. Adamo et al. (2015) found the sparse representation of ECG dataset and proposed a novel and effective signal compression algorithm. Dictionary of waveforms was created from the cardiac signal in the study. It was proven in the study that the proposed k-LiMapS method has high compression rate within a controlled approximation measure. Liu et al. (2016) proposed a dictionary learning algorithm for vector quantization to extract features from ECG signals. The proposed method was also used to eliminate noise from the signals. Chen et al. (2017) presented a novel ECG beat classification algorithm based on a combination of projected and dynamic features. In the study, projected features were derived from a random projection matrix, in which each column was normalized and each row was transformed by Discrete Cosine Transform. The RR intervals were derived as dynamic features. The two kinds of features were classified by the SVM algorithm and better performance

* Corresponding author at: Konya Technical University, Faculty of Engineering and Natural Sciences, Electrical & Electronics Engineering Department, 42250 Konya, Turkey

*E-mail addresses:* mubarstugan@selcuk.edu.tr (M. Barstuğan), rpektatli@selcuk.edu.tr (R. Ceylan).

https://doi.org/10.1016/j.jksuci.2018.11.007
1319-1578/

was obtained according to state-of-art heartbeat classification systems.

In this study, a novel method was proposed to classify the ECG signals. Before the classification stage, the feature selection process was done on ECG signals by DWT, FOS and T-test, Bhattacharyya, Entropy, and Wilcoxon test methods and six different feature subsets were obtained. The obtained feature subsets were classified by the classifier structures, which were formed by AdaBoost, LDA and SVM algorithms. In the proposed method, the first weights of AdaBoost were defined by multiplying sparse representation coefficients which were obtained by K-SVD (K-Singular Value Decomposition) dictionary learning algorithm Aharon et al. (2006). The aim is to implement the classification of ECG signals by AdaBoost, AdaBoost-SVM and AdaBoost-LDA algorithms whose weights were updated by the proposed method and to evaluate the performance of the proposed classifiers. In Huang and Zhang (2014), the proposed method was evaluated on a synthetic dataset and ECG data from a remote diagnosis system. We wanted to implement an AdaBoost classifier based on Dictionary Learning and we tried to prove acceptability of our method on ECG validation dataset (MIT-BIH Arrhythmia).

The paper has four parts. The first part reviews the studies in the literature, which were relevant to dictionary learning based ensemble classifiers. In the second part, updating weights of AdaBoost by using Dictionary Learning method is explained. In the third part, experimental results are presented. The paper concludes with the discussion of the obtained results and suggestions for further research.

## 2. Materials and methods

In this study, AdaBoost, AdaBoost-SVM and AdaBoost-LDA classifiers whose weights were defined by multiplying sparse coefficients were implemented to classify ECG signals.

### 2.1. The dataset used

The used ECG signals were taken from MIT Arrhythmia Database (Physionet, 2017) to evaluate the performance of the proposed method. ECG signals were taken from different patients. The records that were used as the dataset were presented in Table 1.

### 2.2. Pre-processing

In the pre-processing phase, the ECG signals have been preprocessed to detect RR interval. Here, the most important point is considered that all ECG signals are sampled at 360 Hz in Derivation II during recording. Firstly, ECG signals are filtered with low pass and high pass filters before detecting RR intervals of the ECG signal. The cut-off frequencies of low pass and high pass filters are taken as 28 and 0.09 Hz, respectively. Secondly, QRS detection is executed on filtered signal. The detection of RR interval is executed using one of the well-known QRS detection algorithms developed by Friesen et al. (1990). A reason for using this detection algorithm is to give the good detection results for all 3 ECG signal classes when the slope threshold is fixed as 0.45. After detection of R point, the detected RR intervals are arranged as 200 samples (features, data points), which are called as a pattern, for each class (Ceylan et al., 2014).

**Table 1**
The used ECG signals.

| Class | Data files | Number of patterns |
|---|---|---|
| **Normal Sinus Rhythm** | 100, 101, 103, 106, 111, 112, 119, 203 | 1258 |
| **Right Bundle BB** | 118, 207, 212 | 383 |
| **Left Bundle BB** | 109, 207, 214 | 90 |

After the pre-processing phase, three different ECG signal classes that have 200 features were used. The signals, which belong to Normal Sinus Rhythm (N) class, represent the normal beat of the heart. Right Bundle Branch Block (RBBB) signals are a heart block in the electrical conduction system. Left Bundle Branch Block (LBBB) is cardiac conduction abnormality in an electrocardiogram. The used ECG dataset includes 1258 N patterns, 383 RBBB patterns, and 90 LBBB patterns. The features of each pattern can be seen in Fig. 1. Six different feature subsets were created from ECG signals. By creating different feature subsets, it was aimed to find the feature subset which best represents the ECG signals.

### 2.3. Creating of feature subsets

First, to increase the classification accuracy, a feature selection process was applied to ECG signals. Therefore, six different feature subsets were created by DWT, FOS, T-Test, Bhattacharyya, Entropy, and Wilcoxon test methods. By doing feature selection, the features that express the ECG signals better are selected. In this way, the features extracted from different types of ECG signals assist to detect different signal classes. DWT, T-test, Bhattacharyya, Entropy, and Wilcoxon test methods had feature subset which have 15 features, FOS subset had 5 features after feature selection process. The process of creating the feature subsets and classification of these subsets can be seen in Fig. 2.

#### 2.3.1. Creating of feature subset I

The first feature subset was obtained by applying fourth degree DWT to ECG signals. Daubechies DWT was used here. After fourth degree DWT, the approximation coefficients, which have 15 features, were defined as the feature subset I and classified by the proposed method. The implementation of DWT on ECG signals was presented in Fig. 3. As seen in Fig. 3, fourth-degree DWT was applied to ECG signals, and cA4 (last approximation coefficients) was used as feature subset I.

The cA4 coefficients of N, RBBB and LBBB signals can be seen in Fig. 4.

In Fig. 4, cA4 coefficients of one pattern for each class are presented. The obtained feature subset I using cA4 coefficients was used as the new dataset. The dimension of the feature subset I can be seen in Table 2. As seen in Table 2, the pattern number

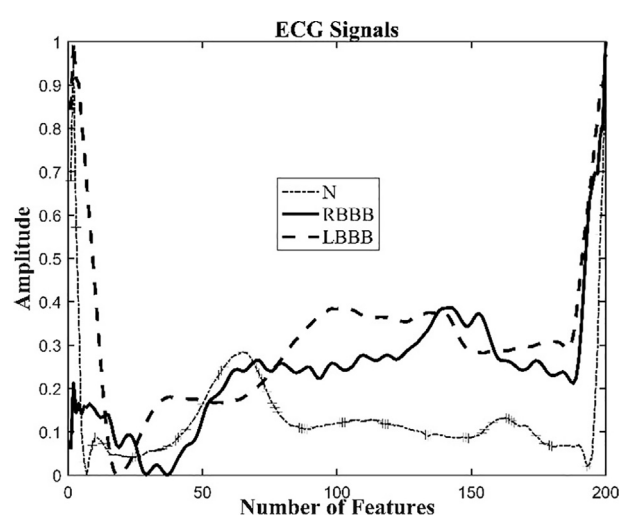


**Fig. 1.** The used ECG patterns.

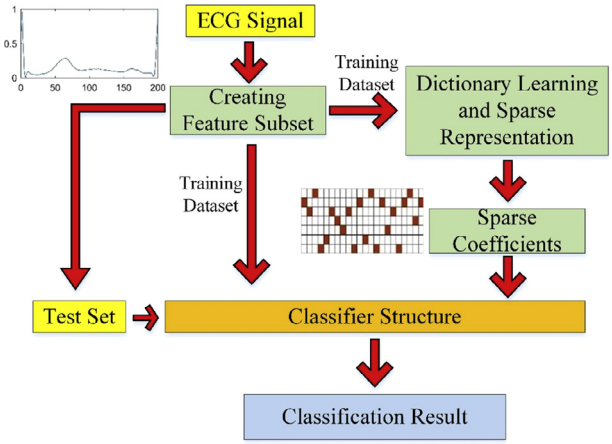


**Fig. 2.** The process of creating feature subsets and classification.

was held the same, only the number of features was reduced by using DWT.

*2.3.2. Creating of feature subset II*

Feature subset II was created by First Order Statistics of each pattern. First Order Statistics include "mean, variance, skewness, kurtosis, shape factor" values. Each of the FOS values was defined as a feature and feature subset II was created as seen in Table 3.

The used statistical methods to obtain the five features were presented as follows Yücelbas et al. (2016), Yücelbaş et al. (2018) (X: Pattern, N: the number of the samples):

$$\text{Mean}: \mu = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1}$$

$$\text{Variance}: \sigma^2 = \sum \frac{X-\mu^2}{N} \tag{2}$$

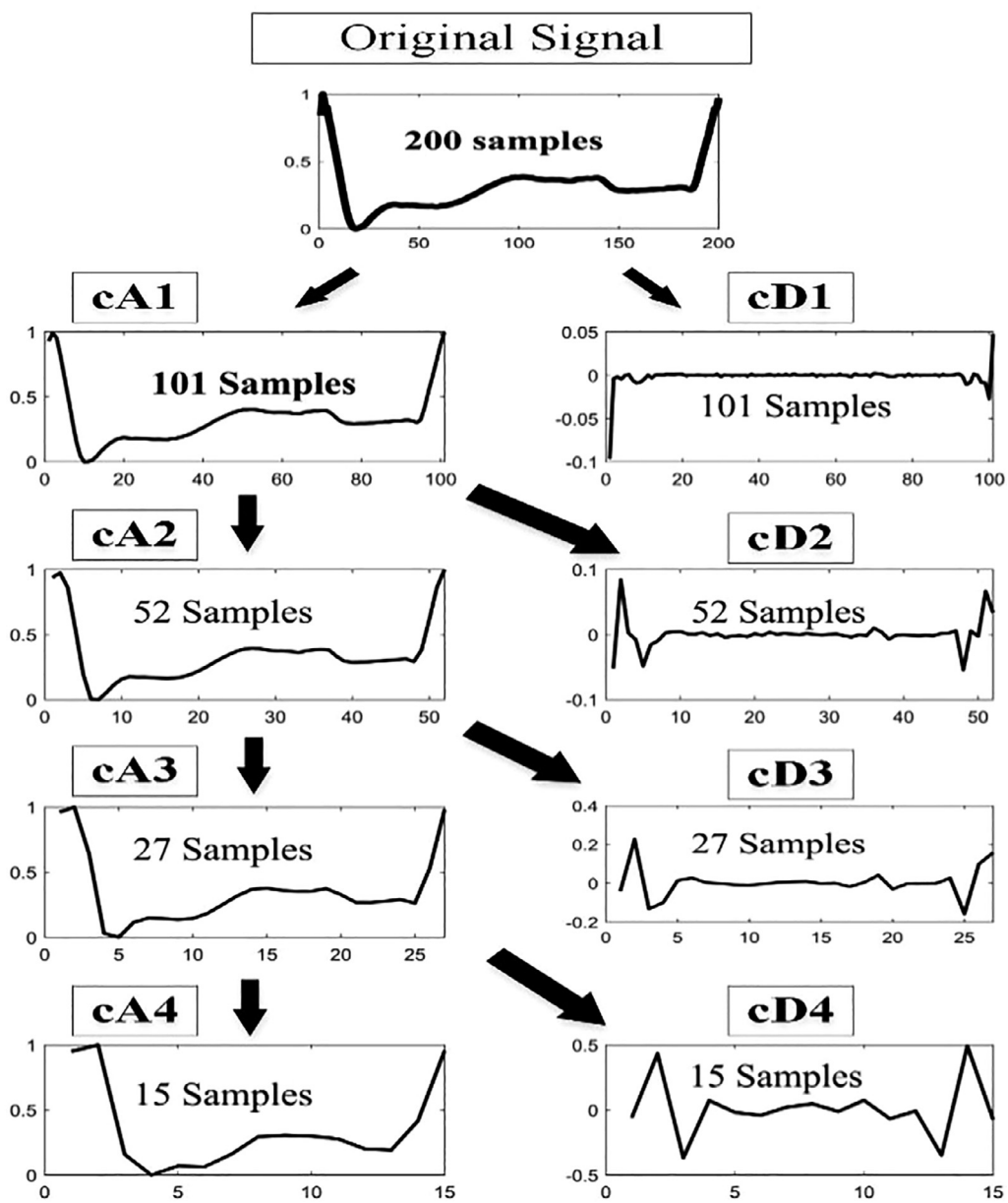


**Fig. 3.** The implementation of fourth-degree DWT on ECG signals.

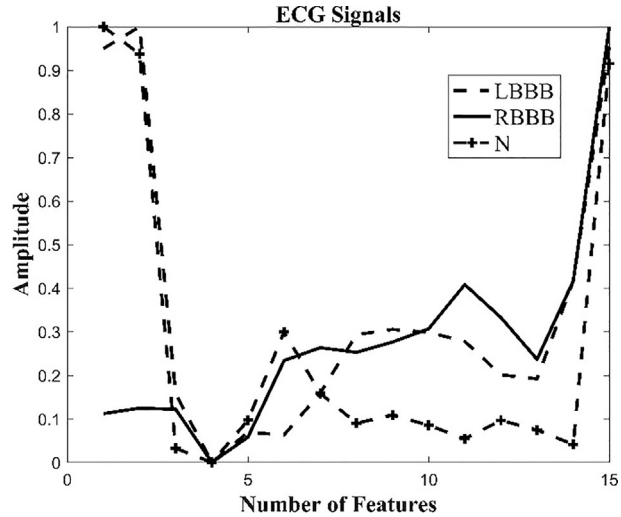


**Fig. 4.** The patterns after the DWT process.

**Table 2**
Feature Subset I.

| Class | Number of features | Number of patterns |
|---|---|---|
| **Normal Sinus Rhythm** | 15 | 1258 |
| **Right Bundle BB** | 15 | 383 |
| **Left Bundle BB** | 15 | 90 |

**Table 3**
Feature Subset II.

| Class | Number of features | Number of patterns |
|---|---|---|
| **Normal Sinus Rhythm** | 5 | 1258 |
| **Right Bundle BB** | 5 | 383 |
| **Left Bundle BB** | 5 | 90 |

$$\text{Skewness}: s = \frac{E(x-\mu)^3}{\sigma^3} \tag{3}$$

$$\text{Kurtosis}: k = \frac{E(x-\mu)^4}{\sigma^4} \tag{4}$$

$$\text{ShapeFactor}: sf = \frac{rms(x)}{\mu(x)} \tag{5}$$

*2.3.3. Creating of feature subset III-VI*

Feature subsets III-VI were defined by T-test, Bhattacharyya, Entropy, and Wilcoxon test methods and all features in ECG signals were ranked. Then the most meaningful first 15 features were obtained by each of the test methods (Haury et al., 2011). The dimensions of Subset III-VI were held the same in accordance with feature subset I.

- *T-test:* T-test is used to determine whatever two sets of data are significantly different from each other.

$$t = \frac{\mu_1 + \mu_2}{\sqrt{\frac{\sigma_1^2+\sigma_2^2}{n_1+n_2}}} \tag{6}$$

where $\mu, \sigma$, and $n$ are sample means, sample standard deviations, and sample sizes of the datasets, respectively.

- *Bhattacharyya:* Bhattacharyya is a method, which measures the similarities between two continuous or discontinuous probability distributions.

$$BD = \frac{1}{4}\ln\left[\frac{1}{4}\left(\frac{\sigma_1^2}{\sigma_2^2}+\frac{\sigma_2^2}{\sigma_1^2}+2\right)\right]\frac{1}{4}\left[\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}\right] \tag{7}$$

- *Entropy:* In statistics, entropy is used to represent the randomness or irregularity of data.

$$e = \frac{1}{2}\left[\left(\frac{\sigma_1^2}{\sigma_2^2}+\frac{\sigma_2^2}{\sigma_1^2}-2\right)+\left(\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}\right)(\mu_1-\mu_2)^2\right] \tag{8}$$

- *Wilcoxon:* Wilcoxon test was developed to test the distributions of two different data classes which are similar or not.

The feature selection process was done by these test methods, and the obtained feature subsets were prepared for classifier structures.

*2.4. Classifier algorithms*

In this study, three different structures were used to classify the created subsets. AdaBoost, AdaBoost-SVM and AdaBoost-LDA structures were combined with Dictionary Learning, and the classification process was done by the proposed method.

*2.4.1. K-SVD algorithm*

In dictionary learning, the aim is to produce a sparse representation of the dataset. If a dataset can be represented by fewer coefficients, the dictionary becomes more effective Tropp and Gilbert (2007). The K-SVD algorithm was proposed by Aharon et al., (2006) as a solution to the Dictionary Learning problem and can be seen in Algorithm 1. K-SVD implements SVD for $k$ times during the update process of dictionary atoms.

**Algorithm 1 – K-SVD**

**Target:** Finding the best dictionary by solving (6) for $\{y_i\}_{i=1}^N$ data

$\min_{D,X}(\{||Y-DX||_F^2\})$ (9)

1: **Input:** D dictionary which was normalized by L2 norm: $||D||_2 = \sqrt{\sum_{i=1}^N D_i^2}$ (10)

2: **for iteration=1**

3: *Sparse Coding Stage*

$\min_{x_i}(\{||y_i - Dx_i||_2^2\})$, $||x_i||_0 \leqslant T_0, i = 1,2,...,N$ (11)

Solve (11) by finding $x_i$ sparse vector for each of $y_i$ sample with a pursuit algorithm.

4: *Dictionary Update Stage*

For each atom in the dictionary:

5: **for k**

6: Define $w_k$ vector:$w_k = \{i|1 \le i \le N\}, x_k^T(i) \neq 0$

7: Compute $E_k$ error matrices:$E_k = Y - \sum_{j\neq k} d_j x_T^j$

8: Obtain$E_k^R$ compressed error matrices by choosing columns from $E_k$ matrices which were defined in $w_k$ vector.

9: Apply SVD to$E_k^R$ matrices:$E_k^R = U\Delta V^T$

10: Update $d_k$ atom by the first column of U. Update$x_k^R$ coefficient vector as$V(:,1) * \Delta(1,1)$

11:**end for k**

12: **end for iteration**

13: **Output:** D (dictionary), x (sparse coefficients)

The OMP algorithm was used to solve Equation 11 in sparse coding stage. OMP finds the closest solution to the problem. In Equation 11, if $T_0$ becomes lower, the solution becomes better (Tropp and Gilbert, 2007).

*2.4.2. Ensemble classifier AdaBoost*

AdaBoost is an ensemble classifier method, which creates a strong classifier by combining weak classifiers. In each iteration, a weak classifier does classification. In continue, depending on the classification error of a weak classifier, the weight of the base learner is updated. A base learner includes several weak classifiers. The best weak classifier is selected as the base learner. The last classification result is obtained by weight voting among the base learners. The AdaBoost algorithm was presented in Algorithm 2 (Kégl, 2009):

**Algorithm 2 – AdaBoost**

1: **Input** *Training dataset* ($x_1$, $y_1$), . . . , ($x_m$,$y_m$); $x_i \in X$, $y_i \in Y = \{-1,+1\}$
$D_1(1,m) = 1/m$ Initialize the weights
2: **for t=1, 2,. . .,T**
3: $\varphi_k$ weak classifiers in the base learner are trained
4: Weak classifier weight is obtained:
arg min $\varepsilon_t = \sum_{i=1}^{m} D_t(i)[y_t \neq \varphi_t(x_i)]$
if $\varepsilon_t > 1/2$; stop the iteration; end
5: Base learner $h_t$ weight $a_t$ is computed$a_t = \frac{1}{2}\log(1-\varepsilon_t/\varepsilon_t)$
6: The weak classifier weights are updated:$D_{t+1} = \frac{D_t(i)\exp(-a_t y_t h_t(x_i))}{Z_t}$
$Z_t$: normalization factor
7: **end for t**
8: **Output** Weight voting result is computed:$H(x) = sign\sum_{t=1}^{T}(a_t h_t(x))$

*2.5. Classifier structures*

In the proposed method, firstly, the initial dictionary was created. The initial dictionary can be created with different methods. In some studies, a part of the data was defined as the initial dictionary. In addition, the initial dictionary can be created randomly or by using some algorithms such as Discrete Cosine Transform, Fourier Transform algorithms. In this study, the initial dictionary was created by Discrete Cosine Transform and normalized with the L2 norm (Alg. 3 – Step 4). By using the training set and initial dictionary, sparse coefficients of training patterns were obtained by Orthogonal Matching Pursuit (OMP) algorithm (Tropp and Gilbert, 2007) (Alg. 3 – Step 5). In the next step, the initial dictionary and sparse coefficients were updated by K-SVD Aharon et al., (2006) algorithm. In AdaBoost algorithm, the weights of the weak classifiers were defined as "1/pattern number". The updated sparse coefficients were multiplied by initial weights, and by doing so, weak classifier weights were updated (Alg. 3 – Step 8). Then, the classification process was done and a classification error was obtained (Alg. 3 – Step 13). The obtained error and sparse coefficients were multiplied, and the weight of first base learner was updated (Alg. 3 – Step 14). In AdaBoost weight update process, the weights of the weak classifiers were updated by using base learner weights (Alg. 2 – Step 6). However, in the proposed method, it was updated by using base learner weights and sparse coefficients (Alg. 3 – Step 15). This process was done for all base learners and the classifier was trained. By using the trained base learners, the test set was classified and the classification performance of the proposed method was obtained. The scheme of this process can be seen in Fig. 5.

Algorithm 3 – DL-AdaBoost (K-SVD-AdaBoost)

1: **Input** Training Dataset ($x_1$, $y_1$), . . . ,($x_m$,$y_m$), $x_i \in X$, $y_i \in Y = \{-1,+1\}$ N: number of base learners, a: number of patterns, M: number of columns in sparse representation coefficient matrices
2: Dataset is normalized with L2 norm
3: Subset is created from dataset
4: Initial dictionary is created by Discrete Cosine Transform
5: Initial dictionary is updated by K-SVD and sparse coefficients ($\psi$) are obtained
6: **for n** = 1:N
7: **for m** = 1:M
8: D(1,a) = $\psi$ (1,m) . (1/a) Define the weights
9: $\varphi_m$ weak classifiers are trained by using the weights
10: The weak classifier which has the least error is selected as a base leaner:
[h{n}, errors] = arg min $h_n = \sum_{i=1}^{m} D(i)[y_i \neq \varphi_i(x_i)]$
11: **end for M**
12: If the least classification error is bigger than 0.5, the iteration is stopped:
if min(errors) > 1/2; stop the iteration; end
13: The base learner error is computed:$\varepsilon_n = \sum(y \neq sign < H_n(x), \psi >)D_n$
14: The base learner weights $a_n$ is computed:$a_n = \frac{1}{2}\log(1-\varepsilon_n/\varepsilon_n)$
15: The weak classifier weights are updated:$D_{n+1} = \frac{D_n \exp((y \neq sign < H_n(x), \psi >))}{Z_n}$
$Z_n$: normalization factor
16: **end for N**
17: The weight voting result is obtained:$H(x) = sign\sum_{n=1}^{N} a_n h_n(x)$

## 3. Experimental results

In this study, AdaBoost, SVM, and LDA algorithms were combined and AdaBoost-LDA, AdaBoost-SVM classifiers were created. Then; DL-AdaBoost, DL-AdaBoost-SVM and DL-AdaBoost-LDA classifiers were created due to the proposed method. To evaluate the effect of the proposed method, the classifiers classified feature subsets. For a fair comparison, each of patterns was used for both of training and test stage by 10-folds cross-validation. AdaBoost, SVM and LDA algorithms are binary classifiers, so One Against One (OAO) (Anthony et al., 2007) method was used to obtain the classification result. Five evaluation metrics (Sensitivity (Sen), Specificity (Spe), Accuracy (Acc), Precision (Pre), Area Under Curve (AUC) values) were used to test the performance of the proposed classifiers. All metrics were calculated according to Eqs. (12–15).

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

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

$$\text{Accuracy} = (\text{TP}+\text{TN})/(\text{TP}+\text{FN}+\text{FP}+\text{TN}) \tag{14}$$

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}) \tag{15}$$

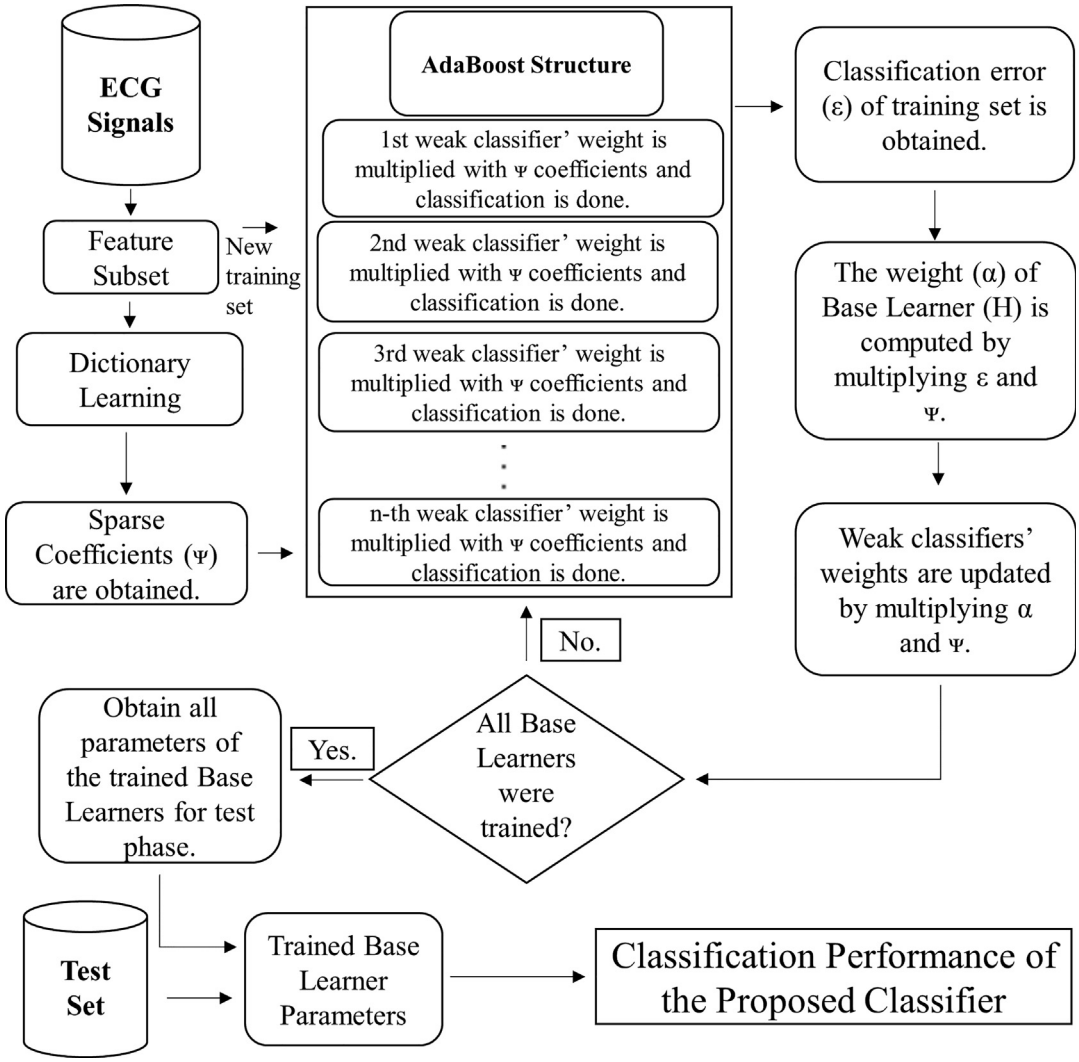


**Fig. 5.** The proposed DL-AdaBoost scheme.

K-SVD parameters were found optimally after experiments. The sparse coefficient number was changed between 3 and 9 by the increment of 2. K-SVD iteration number was changed between 10 and 100 by the increment of 10. Dictionary size was changed between 10 and 50 by the increment of 5. As a result, the optimum sparse coefficient number was found as 3. Optimum iteration number for training K-SVD was found as 10 for feature subsets I, II, III, V, and VI; 70 for feature subset IV. Optimum dictionary size was found as10 for feature subsets I and V; 25 for feature subset II; and 30 for feature subsets III, IV and VI. In Table 4, Sen, Spe, Acc, Pre and AUC values were presented for each implemented classifiers. The evaluation metrics in Table 4 were obtained by taking the mean of 10-folds cross-validation and OAO results.

The experiments were performed on PC. The CPU has Intel i7 4700HQ processor and 12 GB RAM. MATLAB software was used during experiments.

In Table 4, the highlighted accuracy values show that the proposed weight update process increased the classification accuracy. The proposed DL-AdaBoost method could not perform well on feature subset I, and the proposed DL-AdaBoost and DL-AdaBoost-SVM algorithms could not perform well on feature subset II. In the rest of all methods and feature subsets, the weight update process increased the classification accuracy. The best classification accuracy was obtained by the proposed DL-AdaBoost-SVM classifier structure as 99.75% on feature subset I and feature subset VI as seen in Table 4. The obtained results by the proposed method can be seen as graphically in Fig. 6.

As seen in Fig. 6, in ensemble classifiers, the updating weights by multiplying sparse coefficients increases accuracy on the classification of ECG signals. The proposed method has higher accuracy on all experiments; except the experiments on feature subset I by DL-AdaBoost, and the experiments on feature subset II by DL-AdaBoost, DL-AdaBoost-SVM.

The computational costs of the best results obtained in Table 4 were presented in Table 5.

The shortest time was obtained with FOS – DL-AdaBoost-LDA method, but the classification accuracy is the lowest among the best results. The best classification accuracy was obtained with DWT – DL-AdaBoost-SVM and Wilcoxon Test – DL-AdaBoost-SVM as 99.75%. As seen in Table 5, the computational cost of DWT – DL-AdaBoost-SVM is lower than the computational cost of the Wilcoxon Test – DL-AdaBoost-SVM. Therefore, DWT – DL-AdaBoost-SVM method could be chosen as the best method among the six methods.

**Table 4**
The classification results for all subsets.

| Feature Subsets | Evaluation Metrics | | | | |
|---|---|---|---|---|---|
| DWT | Sen | Spe | Acc | Pre | AUC |
| AdaBoost | 99,13 | 99,15 | 98,96 | 96,2 | 0,99 |
| DL-AdaBoost | 98,06 | 98,31 | 98,49 | 98,22 | 0,98 |
| AdaBoost-SVM | 99,39 | 99,79 | **99,47** | 98,32 | 1 |
| DL-AdaBoost-SVM | 98,35 | 99,92 | **99,75** | 99,75 | 0,99 |
| AdaBoost-LDA | 99,04 | 99,84 | **99,35** | 99,24 | 0,99 |
| DL-AdaBoost-LDA | 99,02 | 99,84 | **99,40** | 99,26 | 0,99 |
| **First Order Statistics** | **Sen** | **Spe** | **Acc** | **Pre** | **AUC** |
| AdaBoost | 57,74 | 66,58 | 86,74 | 86,79 | 0,62 |
| DL-AdaBoost | 50,98 | 66,05 | 85,23 | 84,42 | 0,59 |
| AdaBoost-SVM | 97,02 | 97,37 | 95,95 | 83,69 | 0,97 |
| DL-AdaBoost-SVM | 96,85 | 96,59 | 94,93 | 82,03 | 0,97 |
| AdaBoost-LDA | 96,85 | 97,21 | **95,92** | 82,98 | 0,97 |
| DL-AdaBoost-LDA | 97,99 | 97,07 | **96,31** | 82,47 | 0,98 |
| **T-test** | **Sen** | **Spe** | **Acc** | **Pre** | **AUC** |
| AdaBoost | 98,60 | 99,47 | **98,70** | 98,39 | 0,99 |
| DL-AdaBoost | 98,51 | 99,55 | **98,71** | 99,66 | 0,99 |
| AdaBoost-SVM | 98,51 | 99,87 | **98,78** | 98,70 | 0,99 |
| DL-AdaBoost-SVM | 98,41 | 99,26 | **99,09** | 96,96 | 0,99 |
| AdaBoost-LDA | 97,89 | 100,00 | **98,30** | 100,00 | 0,99 |
| DL-AdaBoost-LDA | 97,79 | 99,26 | **98,50** | 99,82 | 0,99 |
| **Bhattacharyya** | **Sen** | **Spe** | **Acc** | **Pre** | **AUC** |
| AdaBoost | 96,90 | 98,07 | **97,72** | 99,00 | 0,97 |
| DL-AdaBoost | 97,72 | 99,50 | **98,27** | 98,55 | 0,99 |
| AdaBoost-SVM | 95,58 | 98,84 | **96,87** | 99,54 | 0,97 |
| DL-AdaBoost-SVM | 96,21 | 99,55 | **97,28** | 90,67 | 0,99 |
| AdaBoost-LDA | 94,62 | 98,86 | **96,21** | 99,61 | 0,97 |
| DL-AdaBoost-LDA | 95,59 | 99,58 | **96,65** | 99,74 | 0,98 |
| **Entropy** | **Sen** | **Spe** | **Acc** | **Pre** | **AUC** |
| AdaBoost | 96,79 | 99,87 | **98,20** | 98,36 | 0,98 |
| DL-AdaBoost | 98,51 | 98,41 | **98,53** | 99,30 | 0,98 |
| AdaBoost-SVM | 94,07 | 99,95 | **96,18** | 99,39 | 0,97 |
| DL-AdaBoost-SVM | 96,56 | 99,23 | **97,38** | 99,73 | 0,98 |
| AdaBoost-LDA | 93,11 | 100,00 | **95,75** | 100,00 | 0,97 |
| DL-AdaBoost-LDA | 95,25 | 99,60 | **96,59** | 99,82 | 0,97 |
| **Wilcoxon Test** | **Sen** | **Spe** | **Acc** | **Pre** | **AUC** |
| AdaBoost | 96,64 | 99,55 | **98,99** | 99,33 | 0,98 |
| DL-AdaBoost | 98,21 | 99,81 | **99,18** | 97,88 | 0,99 |
| AdaBoost-SVM | 99,20 | 99,47 | **99,62** | 98,66 | 0,99 |
| DL-AdaBoost-SVM | 99,74 | 99,55 | **99,75** | 99,61 | 1,00 |
| AdaBoost-LDA | 98,23 | 99,58 | **99,02** | 99,24 | 0,99 |
| DL-AdaBoost-LDA | 98,70 | 99,60 | **99,09** | 99,58 | 0,99 |

## 4. Conclusion

The aim of this paper is to investigate the classification accuracy when AdaBoost and Dictionary Learning were combined and were used as hybrid classifiers. Thus, the proposed method was compared with the other classifiers. 36 different experiments were done in this study. Dictionary Learning was not effective only on six experiments which were not highlighted in Table 4 (The first experiment in DWT section, the first two experiments in First Order Statistics section). In other 30 experiments, the Dictionary Learning increased the classification accuracy. The classification accuracy was over 95% on all experiments with the Dictionary Learning. So, the increment of classification accuracies does not seem high when Dictionary Learning is used. In Table 6, the increment of the classification accuracies by the proposed approach was presented.

As seen in Table 6, the biggest increment is 0.28% on feature subset I by DL-AdaBoost-SVM. The obtained results in this study were compared with literature studies in Table 7.

As seen in Table 7, the proposed method is an effective method and can be used to classify ECG signals. The effectiveness of the proposed method was proven by experimental results. By applying to the different signal classification problem, the performance of the proposed method could be generalized.

## 5. Discussion

Ensemble classifiers method is based on "weight voting" system. After each classification process, the weight of the correctly classified pattern is increased; the weight of the wrongly classified pattern is decreased. During the weight updating process, the weights were updated by multiplying sparse coefficients, and so the number of correctly classified patterns was increased. By increasing the number of correctly classified patterns, the classification performance of the classifier is increased. In addition, to achieve a high classification performance, training of the system with a good dataset is important. Therefore, in this study, six different feature subsets were formed by using DWT, FOS and test methods. The proposed classification approach was evaluated by six different feature subsets. According to computational cost and the highest accuracy metrics, the best feature selection method was found as Discrete Wavelet Transform. By founding the feature subset that best represents the dataset, the classification performance was increased. These experiments show that if a classifier system is trained by a good dataset, it gives higher performance.

The advantage of the proposed method is that a high classification performance can be obtained by using sparse coefficients in Dictionary Learning method. The limitation of the proposed method is that training time takes time because of the number of sparse coefficients. If the number of sparse coefficients higher, the training time becomes higher.

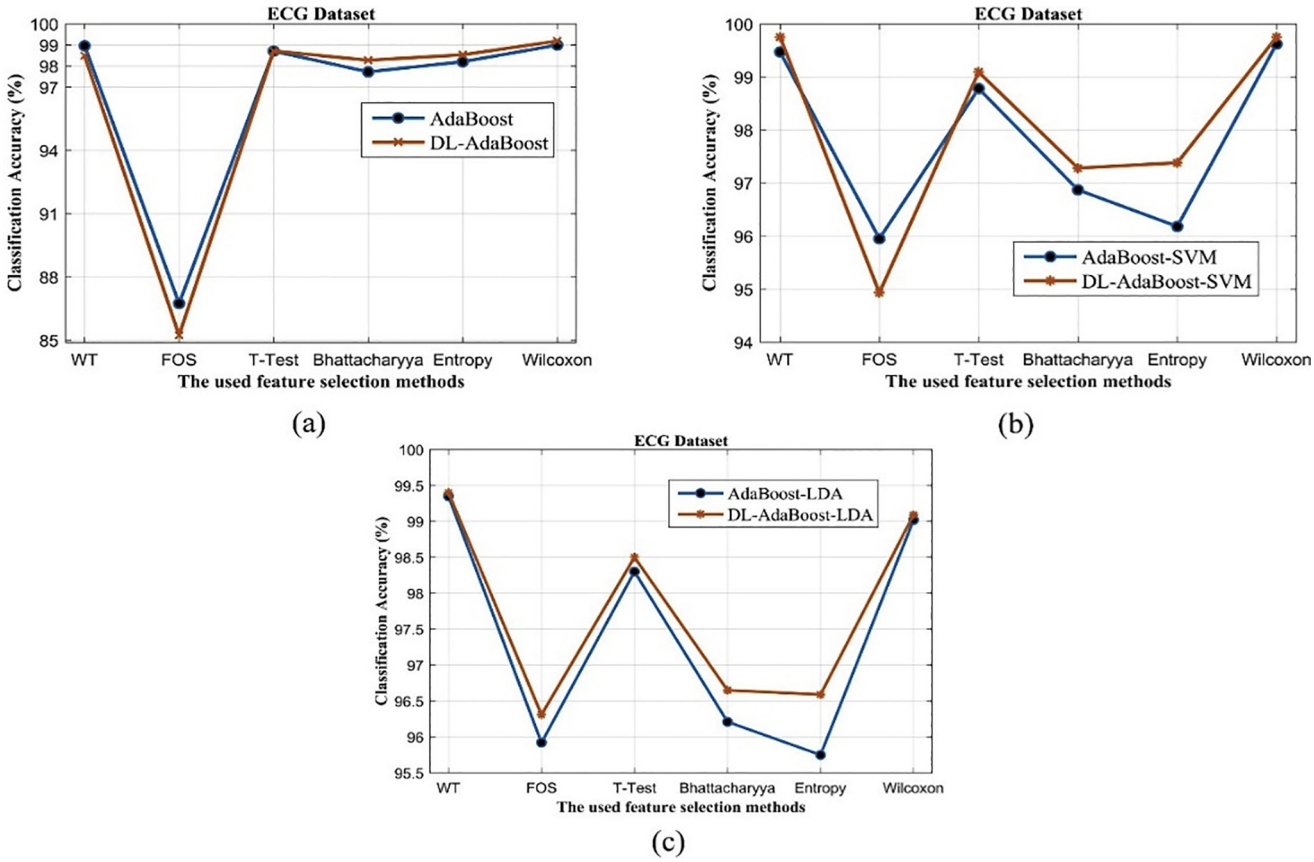


**Fig. 6.** The classification results for all feature subsets by (a) AdaBoost and DL-AdaBoost (b) AdaBoost-SVM and DL-AdaBoost-SVM (c) AdaBoost-LDA and DL-AdaBoost-LDA.

**Table 5**
Computational costs of the best methods in Table 4.

| Method | CPU time (sec) | Processed Lines |
|---|---|---|
| **DWT – DL-AdaBoost-SVM** | **1884.94** | **52658401** |
| FOS – DL-AdaBoost-LDA | 100.89 | 680766 |
| Wilcoxon Test – DL-AdaBoost-SVM | 2918.31 | 93732293 |

**Table 6**
The increment of classification accuracies when the Dictionary Learning is used.

| Classifier | AdaBoost | DL-AdaBoost | AdaBoost-SVM | DL-AdaBoost-SVM | AdaBoost-LDA | DL-AdaBoost-LDA |
|---|---|---|---|---|---|---|
| Subset | Subset VI | | Subset I | | Subset I | |
| Classification Accuracy (%) | 98.99→99.18 | | 99.47 →99.75 | | 99.35 →99.40 | |
| Increment (%) | **0.19** | | **0.28** | | **0.05** | |

**Table 7**
Comparison with the recent results.

| Author | Year | Method | Performance (%) |
|---|---|---|---|
| Liu et al. (2016) | 2017 | Dictionary Learning-Vector Quantization | 94.6 |
| Chen et al. (2017) | 2016 | Feature Extraction-SVM | 98.46 |
| Li et al. (2017) | 2016 | General Regression Neural Network | 95 |
| Li et al. (2016) | 2016 | Feature Extraction-SVM | 98.65 |
| Raj and Ray (2017) | 2016 | Hybrid PSO-SVM | 98.82 |
| Sharma and Ray (2016) | 2016 | Multi-class SVM with RBF Kernel | 99.51 |
| Raj et al. (2016) | 2016 | DOST+SVM-PSO | 99.18 |
| Elhaj et al. (2016) | 2016 | SVM-RBF | 98.91 |
| Rivera and Rodriguez (2017) | 2017 | Adaptive Neuro-Fuzzy | 98.38 |
| Ceylan (2018) | 2018 | K-SVD-ANN | 98.74 |
| Oh et al. (2018) | 2018 | CNN and LSTM combination | 98.10 |
| Yildirim (2018) | 2018 | DBLSTM-WS | 99.39 |
| This Study | 2018 | Dictionary Learning-AdaBoost-SVM | 99.75 |

## Acknowledgment

This work is supported by the Coordinatorship of Selcuk University's Scientific Research Projects.

## References

Adamo, A., Grossi, G., Lanzarotti, R., Lin, J., 2015. ECG compression retaining the best natural basis k-coefficients via sparse decomposition. Biomed. Signal Process. Control 15, 11–17.

Aharon, M., Elad, M., Bruckstein, A., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. 54, 4311.

G. Anthony, H. Gregg, M. Tshilidzi, Image classification using SVMs: one-against-one vs one-against-all, arXiv preprint arXiv: 0711.2914, 2007.

Ceylan, R., 2018. The effect of feature extraction based on dictionary learning on ECG signal classification. Int. J. Intellig. Syst. Appl. Eng. 6, 40–46.

Ceylan, R., Özbay, Y., Karlik, B., 2014. comparison of type-2 fuzzy clustering-based cascade classifier models for ECG arrhythmias. Biomed. Eng.: Appl. Basis Commun. 26, 1450075.

Chen, S., Hua, W., Li, Z., Li, J., Gao, X., 2017. Heartbeat classification using projected and dynamic features of ECG signal. Biomed. Signal Process. Control 31, 165–173.

Elhaj, F.A., Salim, N., Harris, A.R., Swee, T.T., Ahmed, T., 2016. Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals. Comput. Method. Program. Biomed. 127, 52–63.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55, 119–139.

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

Haury, A.-C., Gestraud, P., Vert, J.-P., 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE 6. e28210.

https://www.physionet.org/physiobank/database/mitdb/. 2017.

Huang, K., Zhang, L.-Q., 2014. Semisupervised sparse multilinear discriminant analysis. J. Comput. Sci. Technol. 29, 1058–1071.

B. Kégl, Introduction to AdaBoost. 2009.

Li, P., Wang, Y., He, J., Wang, L., Tian, Y., Zhou, T.-S., Li, T., Li, J.-S., 2017. High-performance personalized heartbeat classification model for long-term ECG signal. IEEE Trans. Biomed. Eng. 64, 78–86.

Li, H., Yuan, D., Wang, Y., Cui, D., Cao, L., 2016. Arrhythmia classification based on multi-domain feature extraction for an ECG recognition system. Sensors 16, 1744.

Liu, T., Si, Y., Wen, D., Zang, M., Lang, L., 2016. Dictionary learning for VQ feature extraction in ECG beats classification. Expert Syst. Appl. 53, 129–137.

Oh, S.L., Ng, E.Y., San Tan, R., Acharya, U.R., 2018. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. Comput. Biol. Med.

Raj, S., Ray, K.C., 2017. ECG signal analysis using DCT-based DOST and PSO optimized SVM. IEEE Trans. Instrum. Meas. 66, 470–478.

Raj, S., Ray, K.C., Shankar, O., 2016. Cardiac arrhythmia beat classification using DOST and PSO tuned SVM. Comput. Method. Program. Biomed. 136, 163–177.

J.N. Rivera, K.C. Rodriguez, ECG Classification using Adaptive Neuro-Fuzzy Inference System.

Sharma, P., Ray, K.C., 2016. Efficient methodology for electrocardiogram beat classification. IET Signal Proc. 10, 825–832.

Tropp, J.A., Gilbert, A.C., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory 53, 4655–4666.

Wang, B., Pineau, J., 2016. Generalized dictionary for multitask learning with boosting. IJCAI, 2097–2103.

Yildirim, Ö., 2018. A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification. Comput. Biol. Med. 96, 189–202.

Yücelbas, C., Yucelbas, S., Ozsen, S., Tezel, G., Kuccukturk, S., Yosunkaya, S., 2016. Detection of sleep spindles in sleep EEG by using the PSD methods. Indian J. Sci. Technol. 9.

Yücelbaş, C., Yücelbaş, Ş., Özşen, S., Tezel, G., Küççüktürk, S., Yosunkaya, Ş., 2018. Automatic detection of sleep spindles with the use of STFT, EMD and DWT methods. Neural Comput. Appl. 29, 17–33.

Zhang, L., Zhou, W.-D., 2011. Sparse ensembles using weighted combination methods based on linear programming. Pattern Recogn. 44, 97–106.