Decision tree ensembles in biomedical time-series classification

Alan Jović*, Karla Brkić*, and Nikola Bogunović*

*Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, 10000 Zagreb, Croatia
alan.jovic@fer.hr, karla.brkic@fer.hr, nikola.bogunovic@fer.hr

Abstract. There are numerous classification methods developed in the field of machine learning. Some of these methods, such as artificial neural networks and support vector machines, are used extensively in biomedical time-series classification. Other methods have been used less often for no apparent reason. The aim of this work is to examine the applicability of decision tree ensembles as strong and practical classification algorithms in biomedical domain. We consider four common decision tree ensembles: AdaBoost.M1+C4.5, MultiBoost+C4.5, random forest, and rotation forest. The decision tree ensembles are compared with SMO-based support vector machines classifiers (linear, squared polynomial, and radial kernel) on three distinct biomedical time-series datasets. For evaluation purposes, 10x10-fold cross-validation is used and the classifiers are measured in terms of sensitivity, specificity, and speed of model construction. The classifiers are compared in terms of statistically significant wins-losses-ties on the three datasets. We show that the overall results favor decision tree ensembles over SMO-based support vector machines. Preliminary results suggest that AdaBoost.M1 and MultiBoost are the best of the examined classifiers, with no statistically significant difference between them. These results should encourage the use of decision tree ensembles in biomedical time-series datasets where optimal model accuracy is sought.

1 Introduction

Biomedical time-series (BTS) are series of measurements taken from a complex biological system with the basic purpose of diagnosis and treatment of the disorders present in the system. The most common types of BTS measured today are: heart rhythm, electrocardiogram (ECG), electroencephalogram (EEG), electromyogram (EMG), pulse oxymetry, and others [1]. Any analysis of BTS needs to include specific measures (features) that describe it in a way that is the most suitable for discerning different disorder patterns. The cardinality of the feature space of almost any type of BTS is infinite. A "good" set of selected features is the one that would allow the researcher to differentiate the patterns present in the series with ease. Obtaining such a set is highly dependent on the following: type of BTS, types of analyzed patterns, noise present in the BTS, and availability of patient data. In general, the researchers do not agree on the optimal set of features for a specific disorder, and clinical guidelines provide only limited recommendations about the use of particular features [2].

Arguably, the most common type of BTS analysis is classification, wherein the researcher seeks a model that would allow him to accurately classify two or more disorders present in the BTS. Many approaches to classification of BTS have been presented in literature. Some of the more common ones include various types of artificial neural networks (ANN) [3,4] and support vector machines (SVM) [5,6]. Other methods such as Bayesian networks [7] and decision trees (CART) [8] are used occasionally. From the perspective of both the algorithm's accuracy and speed, it is mostly unclear why various forms of decision trees are not used more often for classification of BTS. One of the possible reasons may be that the researchers regard only a single decision tree such as C4.5 or CART (due to interpretability), which is not strong enough to compare to the classification results of ANN or SVM. Another reason is that ANN and SVM are theoretically more well-founded. Nevertheless, the researchers in the field of BTS analysis might not be aware of more recent development in decision tree based classifiers.

In this work, we focus on ensembles of decision trees classifiers and compare them with the SVM classifiers. Some of the known ensembles include AdaBoost, MultiBoost, random forest, and rotation forest. Decision tree ensembles tend to produce very accurate results on a variety of datasets due to the reduction in both bias and variance component of the generalization error of the base classifier [9]. Previous work indicated that some of the decision tree ensembles (AdaBoost.M1+C4.5 and random forest) may have advantages over ANN and SVM classifiers in classification of heart rhythm time-series in terms of speed and accuracy [10]. SVM classifiers are known to have some advantages over ANN, particularly in terms of accuracy and overfitting avoidance [11]. Therefore, in this paper, we focus on SVM and reserve the consideration of ANN algorithms for future work. This paper aims to determine: 1) whether decision tree ensembles are comparable to or better than SVM algorithms in terms of accuracy and speed of BTS classification models, and 2) which of the inspected decision tree ensembles gives the best results for BTS classification.

We will restrict the work presented in this paper to multiclass classification of a single output class categorical attribute (disorder type), with input attributes being BTS features of either categorical or numerical type. This covers most of the BTS classification datasets. Also, the restriction is on classification methods that have models with no clear interpretation, with the aim of maximum model accuracy. The analysis will be empirical, as the algorithms will be compared on three distinct datasets. The first one is the well-known "Arrhythmia" dataset from UCI repository [12] that contains 16 categories of patient disorders. The dataset is mostly based on ECG time-series with some additional patient information. The second and third datasets are based only on features of heart rate variability extracted from freely available PhysioNet databases [13] with different classification goals.

The structure of this paper is as follows. In Section 2, we describe the employed machine learning algorithms. An overview of the datasets and the classification procedure are given in Section 3. The results of the comparison between the constructed classifiers are presented and discussed in Section 4. Conclusion is given in Section 5.

2 Classification methods

2.1 AdaBoost.M1+C4.5

AdaBoost.M1 (AB) is a well-known algorithm for boosting weak classifiers [14]. AB is a member of a broader family of iterative machine learning algorithms that build the final classifier through a finite series of improvements to the classifier. The idea of the AB algorithm is to penalize the instances in the training set that are correctly classified by the classifier. The penalized instances then have a smaller chance to be reselected for the training set. The algorithm focuses on the more problematic instances in each successive step.

Let *K* be the number of successive steps of the AB algorithm. In the first step, the algorithm selects *N* (*N* is the number of instances in the training set) instances to form the first training set by randomly taking instances with equal chance from the initial set with replacement (bootstrap method). Each instance may be selected more than once or may not be selected at all. The algorithm then trains the base classifier and classifies the instances. The instances that are correctly classified receive the penalty to their weight for the next step:

$$w_j^{i+1} := w_j^i \frac{e_i}{1-e_i}, \quad e_i = \sum_{j=1}^{r} w_j^i e(x_j), \quad e(x_j) = \begin{cases} 1, f_i^{Alg}(x_j) = y_j \\ 0, f_i^{Alg}(x_j) \neq y_j \end{cases} \qquad (1),$$

where $f_i^{Alg}(x_j)$ denotes the i-th classifier built by the classification algorithm Alg (e.g. C4.5). Additional modifications of the weights are possible in cases where error exceeds 50% or drops to 0. The algorithm terminates after a number of successive steps *K* is reached. A weight is contributed to each constructed classifier. In the testing phase, each classifier in row gives a prognosis for the target class. Each time a target class is selected, its weight is increased depending on the weight of the classifiers. Finally, voting is performed that selects the target class with the highest weight.

The reason why AB is so successful is because it significantly lowers both the classifier variance and bias errors [9]. High variance error is typical for most of the decision tree algorithms, including C4.5. Originally, AdaBoost.M1 used a very simple decision tree, decision stump, as base classifier. Some researchers noticed that better classification results might be obtained if C4.5 is used as the base classifier instead of the stump. C4.5 can deal with weights associated to instances. AB has been shown to significantly improve the results of the basic C4.5 algorithm on a variety of datasets, including biomedical data [9,15].

2.2 MultiBoost+C4.5

MultiBoost (MB) is regarded as an extension to AdaBoost that combines the AB algorithm with the wagging procedure, which is itself extension of the basic bagging method [16]. Instead of *K* single classifiers used by the AB algorithm, MB constructs a number of sub-committees consisting of a number of trees. Each sub-committee has its own specific iteration $I_k \leq K, \sum_k I_k = K$ in which it terminates. Sub-committee is formed by AB using wagging instead of bootstrap. Wagging works by setting random weights of instances to those drawn from an approximation of the continuous Poisson distribution. After the weights are assigned, the vector of weights is always standardized to sum to *N*. All instances in the training set are used to train the base classifier using the designated weights. The weights are corrected in each subsequent step of constructing the sub-committee by using (1). All other steps of MB are equal to the AB algorithm, including the testing phase. Using C4.5 as the base classifier for MB is straightforward, as C4.5 handles weights associated to instances. Wagging is shown to be particularly successful in reducing the variance error. Therefore, the combination of wagging and AB can, in principle, lead to better results. MB can also be parallelized at the sub-committee level.

2.3 Random forest

Random forest (RF) is a decision tree ensemble learner developed by Breiman [17]. RF supports classification, regression, feature selection, prototyping, and other data mining methods. Decision trees that compose the forest are constructed by choosing their splitting attributes from a random subset of *k* attributes at each internal node. The best split is taken among these randomly chosen attributes and the trees are built without pruning, as opposed to C4.5. The quality of the split at an attribute is determined by its Gini impurity index. RF avoids overfitting due to two sources of randomness - the aforementioned random attribute subset selection and bootstrap training set sampling. Breiman has shown that if one constructs the forest consisting of a large enough number of such decision trees, the overall classification error will be minimized and the accuracy will reach a plateau. RF is widely used in various classification problems, especially in domains with larger numbers of attributes and instances, because of its high speed and accuracy [17].

2.4 Rotation forest

Rotation forest (RTF) is a more recent decision tree ensemble method proposed by Rodriguez et al. [18]. The ensemble is capable of both classification and regression, depending on the base classifier. In most applications, C4.5 algorithm is used as the base learner. Algorithm focuses on presenting transformed data to the classifier by using a projection filter. The most common projection filter and the one that has been shown to be the main factor for the success of the ensemble is the principal component filter [19].

Let the number of base classifiers be given as *K*. In order to create the training set for each base classifier, the instances are first sampled using the bootstrap method. Next, the feature set is randomly split into *M* subsets and principal component analysis is applied to each subset. All of the eigenvectors are retained as the new features in order to preserve the variance in the data. The idea why these *M* data transformations are performed is to encourage simultaneously individual accuracy and diversity of classifiers within the ensemble, as this is the most important precondition for a successful ensemble [17]. Diversity is achieved through random splitting of the feature set, and accuracy is sought by retaining all the principal components.

2.5 Support vector machines

Support vector machines (SVM) is a kernel based machine learning family of methods that are used to accurately classify both linearly separable and linearly inseparable data [20]. The basic idea when the data is not linearly separable is to transform them to a higher dimensional space by using a transformation kernel function. In this new space the samples can usually be classified with higher accuracy. Many types of kernel functions have been developed, with the most used ones being polynomial and radial-based.

In this work, three types of SVM are considered: linear SVM, squared polynomial SVM, and radial-based SVM. As the learning method, sequential minimal optimization (SMO) type algorithm will be used. The proposed algorithm efficiently resolves quadratic programming optimization problem that arises when determining the maximum margin hyperplane of the support vector machines classifier. Original work on SMO by Platt was later optimized by Keerthi et al. [21], and this optimization, which is implemented in Weka platform [22], will be used. We also considered the LIBSVM [23] implementation of *C*-SVC in Java as proposed by Fan et al. [24]; however training times were an order of magnitude higher than the Keerthi's method with no improvement in accuracy of the models.

Because SMO is a binary classification algorithm, for multiclass classification purposes it is adapted such that it performs $n*(n-1)/2$ binary classifications. The SVM algorithm is parametric and deterministic. The most significant parameters are the cost of the margin and the radial kernel parameter gamma (γ).

3 Datasets and evaluation specifics

3.1 Datasets

The first considered dataset is the "Arrhythmia" dataset from UCI repository [14]. Arrhythmia dataset contains a total of 452 instances of 12-lead ECG measurements. 275 features are extracted from each ECG and additional four patient features are taken into consideration (age, sex, height, weight) to a total of 279 predictive attributes. ECG features include mostly morphological characteristics of observed ECG waves and wave to wave interval durations. Most features are numerical (206), and the rest are categorical or binary (73). There is a single output attribute (ECG class), with 16 possible types, out of which 13 are actually present in the dataset. The majority of the dataset is covered by examples of normal ECGs. Due to the lack of data for some of the ECG disorders, this dataset is considered by some to be difficult for classification, with reported results achieving only 62% total classification accuracy [25].

The second dataset is obtained by extracting features from heart rate variability (HRV) records from two MIT-BIH databases (Arrhythmia and Supraventricular Arrhythmia), available from PhysioNet [13]. For feature extraction, we used the HRVFrame framework of Jović and Bogunović [26]. We extracted a total of 230 numerical features from 125 patient records from both databases. The features included linear time domain, frequency domain, time-frequency, and a large number of nonlinear features. A total of 8843 instances were obtained for time-segments of 20 s, which is known to be near-optimal segment duration for arrhythmia detection [6,10]. The goal was to classify 9 types of commonly occurring heart rhythm patterns found in the databases. To our knowledge, there exists no previously published research that used these two databases.

The third dataset is obtained by extracting HRV features from six MIT-BIH databases, which include: MIT-BIH Normal Sinus Rhythm, Normal Sinus Rhythm RR Interval, MIT-BIH Arrhythmia, MIT-BIH Supraventricular Arrhythmia, BIDMC Congestive Heart Failure, and Congestive Heart Failure RR-interval [13]. A total of 3317 instances from 237 records were obtained for time segments of 5 min with intention of finding potentially accurate models for distinction of the three patient groups: healthy persons, arrhythmic patients, and congestive heart failure (CHF, of different severity level). For feature extraction, we also employed the HRVFrame framework of Jović and Bogunović [26] and extracted a total of 237 numerical features. The features included linear time domain, frequency domain, time-frequency, and a large number of nonlinear features. Previously reported accuracy on this dataset for four-class classification (supraventricular arrhythmia was considered as a separate class of arrhythmia) was 72% [27]. Distribution of instances for all three datasets is shown in Table 1. More on these disorders can be found in [28].

3.2 Evaluation specifics

For evaluation of the classifiers on each dataset, we use 10x10-fold cross-validation. Evaluation measures used are standard in BTS analyses: sensitivity (SENS), and specificity (SPEC):

$$SENS = \frac{TP}{TP+FN},\ SPEC = \frac{TN}{TN+FP} \qquad (2),$$

where TP, TN, FP, and FN are the numbers of: true positives, true negatives, false positives, and false negatives, respectively. For multiclass case, these measures can be obtained from the confusion matrix by comparing numbers of instances for each class in the matrix against instances of all the other classes. The reported values have been weighted and averaged among classes.

Parameters of the algorithms were modified in order to obtain the best possible result using systematic approach on the first 10-fold iteration. The other nine 10-fold

Table 1. Distribution of disorders/rhythm patterns in the two analyzed datasets

Dataset (instance count)	Disorder (instance count)
Arrhythmia (452)	Normal (245), CAD (44), Old anterior MI (15), Old inferior MI (15), Tachycardia (13), Bradycardia (25), PVC (3), PAC (2), LBBB (9), RBBB (50), LV hypertrophy (4), AFIB or AFL (5), Other (22)
HRV (8843)	NSR (4121), PAC (1065), PVC (1466), AFIB (749), VBI (375), VTR (299), ABI (272), ATR(178), PACE(318)
CHF (3317)	Healthy (1182), Arrhythmic (1328), Congestive heart failure (827)

iterations are used for obtaining classification results. Possible combinations of parameters were evaluated by increasing their values step-wise in Weka until an optimal setting (with respect to maximum model sensitivity) on the first 10-fold iteration was found. The searching set for each parameter differed, thus, e.g. number of iterations of AB started from 10 and continued to 100 with the step of 10, while cost parameter C for SVM started at 0.01 and continued to 2048 but with a nonlinear step increase.

For the analyzed datasets, only the best parameters for the SVM classifiers differed. For other classifiers, the optimal parameters on all three datasets were: AB: 40 iter., C4.5 pruning conf. = 0.4; MB: 40 iter., 5 sub-committees, C4.5 pruning conf. = 0.4; RF: 100 trees, max. depth = 20; RTF: 30 iter., 8 max. and 8 min. group members. SVM: Arrhythmia dataset (SVM lin.: C=1; SVM sq.: C=0.03, included first order, SVM rad.: C=5, γ=0.15), HRV dataset (SVM lin.: C=1; SVM sq.: C=0.03, included first order, SVM rad.: C=128, γ=0.01), CHF dataset (SVM lin.: C=10; SVM sq.: C=0.15, included first order, SVM rad.: C=512, γ=0.15). Variations in parameter values would probably be greater if the datasets had more diverse feature counts (around 200 features were present in all three datasets).

4 Results

The algorithms were compared by their mean values and standard deviations for each evaluation measure obtained on nine 10-fold cross-validation iterations. Results for the three datasets are shown in Fig. 1. It is noticeable that the boosting algorithms perform favorably to the other algorithms. Also, if one disregards the apparent failure of RF on the first dataset (probably because of too few examples for the random trees to learn from), all decision tree ensembles compare favorably to SVM algorithms, both for sensitivity and for specificity. To confirm this result statistically, win-loss-tie comparison of the algorithms based on sensitivity is presented in Table 2. Statistically significant wins and losses were obtained using two-sided paired Student t-test on the mean results for the nine 10-fold iterations, with significance level $\alpha = 0.05$. Apparently, SVM algorithms lose to decision tree ensembles on all three datasets (except to RF on the Arrhythmia dataset). Arguably, radial SVM gave the best results among the SVM SMO classifiers. RF and RTF seem to have similar results on average, and they lose to both AB and MB. There are no significant differences between the two best ensembles.

The failure of SVM may be surprising, but it is probably due to the characteristics of the datasets that are common in this domain. All three datasets are multiclass and the samples are not linearly separable. Extending SVM to multiclass case is not straightforward [29], and the implemented SMO algorithms used pair-wise classification. It would be interesting to see if one-vs.-all strategy would lead to better results.

In Table 3, average times needed for classifier construction for the three datasets are shown. Random forest is the fastest algorithm overall. AdaBoost.M1+C4.5 and MultiBoost+C4.5 have reasonably satisfactory model construction times, although slower than most of the other algorithms.

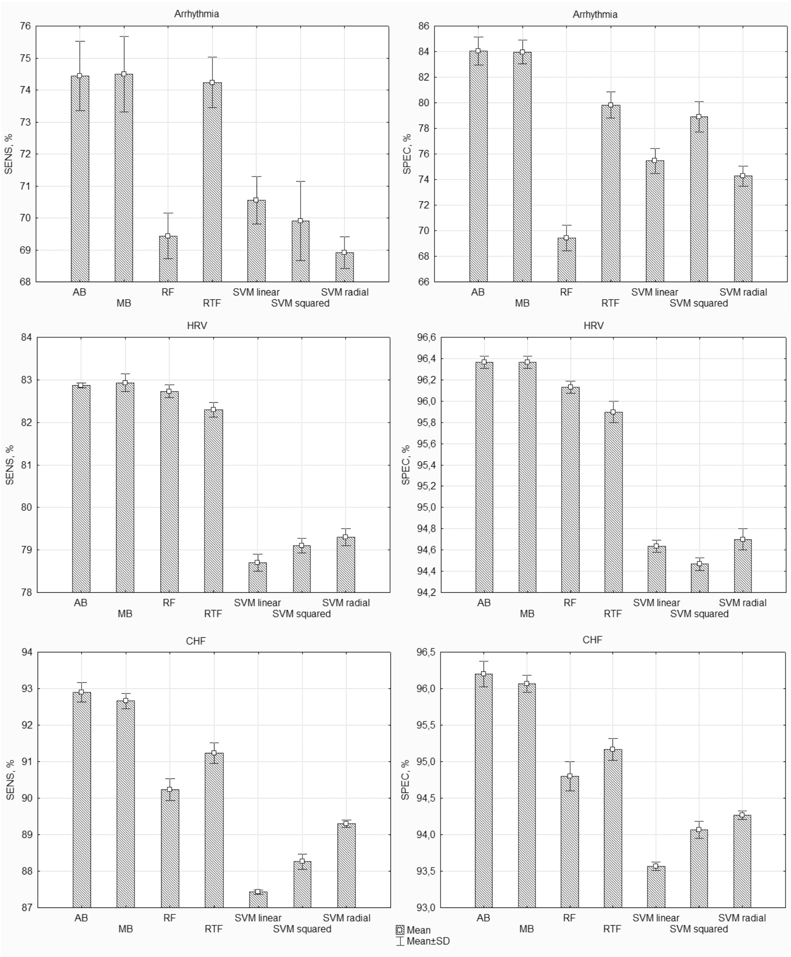

Fig. 1. Sensitivity and specificity of the classifiers' models for the three datasets

Table 2. Win/loss/tie comparison of the algorithms based on sensitivity for the three datasets, Student's paired t-test, $\alpha = 0.05$

vs.	AB	MB	RF	RTF	SVM linear	SVM squared	SVM radial
AB	-	0/0/3	2/0/1	2/0/1	3/0/0	3/0/0	3/0/0
MB	0/0/3	-	2/0/1	2/0/1	3/0/0	3/0/0	3/0/0
RF	0/2/1	0/2/1	-	1/1/1	2/1/0	2/0/1	2/0/1
RTF	0/2/1	0/2/1	1/1/1	-	3/0/0	3/0/0	3/0/0
SVM linear	0/3/0	0/3/0	1/2/0	0/3/0	-	1/2/0	1/1/1
SVM squared	0/3/0	0/3/0	0/2/1	0/3/0	2/1/0	-	0/1/2
SVM radial	0/3/0	0/3/0	0/2/1	0/3/0	1/1/1	1/0/2	-

Table 3. Average classification model construction times (in seconds) for the three datasets

Dataset	AB	MB	RF	RTF	SVM linear	SVM squared	SVM radial
Arrhythmia	21.8±0.1	22.4±0.3	2.8±0.2	48.4±2.6	0.7±0.1	1.0±0.1	1.0±0.1
HRV	675.6±8.9	616.1±1.8	29.8±1.0	825.3±3.8	135.4±13.3	301.7±4.4	370.3±3.0
CHF	177.3±2.8	177.0±4.2	8.7±0.1	192.5±4.2	77.2±3.4	63.3±0.2	81.9±3.6

5 Conclusion

This work examined the use of decision tree ensembles in biomedical time-series classification. These algorithms are shown to be accurate and fast, as they construct diverse classifiers in little time, and vote strongly for the target class.

The analysis has been limited to only three biomedical time-series datasets, all three related to cardiac disorders. The preliminary results suggest that the ensembles compare favorably to SVM-based classifiers. Future work should inspect a larger number of cardiac disorders datasets as well as other biomedical time-series datasets to determine whether the analyzed methods achieve similar results. The results presented in this work clearly support the use of decision tree ensembles in biomedical time-series classification. In particular, AdaBoost.M1 and MultiBoost algorithms applied to C4.5 decision tree seem to be the most accurate with satisfactory model construction times.

Acknowledgements. This research has been partially funded by the University of Zagreb Development Fund.

References

1. Glass, L.: Synchronization and rhythmic processes in physiology. Nature 410, 277–284 (2001)
2. Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology: Heart rate variability guidelines: Standards of measurement, physiological interpretation, and clinical use. Eur. Heart J. 17, 354–381 (1996)
3. Acharya, R.U., Kumar, A., Bhat, P.S., Lim, C.M., lyengar, S.S., Kannathal, N., Krishnan, S.M.: Classification of cardiac abnormalities using heart rate signals. Med. Biol. Eng. Comput. 42, 288–293 (2004)
4. Manis, G., Nikolopoulos, S., Alexandridi, A., Davos, C.: Assessment of the classification capability of prediction and approximation methods for HRV analysis. Comp. Biol. Med. 37, 642–654 (2007)
5. Osowski, S., Hoai, L.T., Markiewicz, T.: Support vector machine-based expert system for reliable heartbeat recognition. IEEE Trans. Biomed. Eng. 51, 582–589 (2004)
6. Asl, B.M., Setarehdan, S.K., Mohebbi, M.: Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. Artif. Intell. Med. 44(1), 51–64 (2008)
7. Soman, T., Bobbie, P.O.: Classification of Arrhythmia Using Machine Learning Techniques. WSEAS Trans. Comp. 4(6), 548–552 (2005)

8. Pecchia, L., Melillo, P., Sansone, M., Bracale, M.: Discrimination power of short-term heart rate variability measures for CHF assessment. IEEE Trans. Inf. Technol. Biomed. 15(1), 40–46 (2011)
9. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. In: 14th Int. Conf. Mach. Learn., pp. 322–330. Morgan Kaufmann, San Francisco (1997)
10. Jović, A., Bogunović, N.: Evaluating and comparing performance of feature combinations of heart rate variability measures for cardiac rhythm classification. Biomed. Signal Process. Control, in press, doi:10.1016/j.bspc.2011.10.001
11. Lisboa, P.J., Vellido, A., Wong, H.: Bias reduction in skewed binary classification with Bayesian neural networks. Neural Networks 13, 407–410 (2000)
12. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
13. PhysioNet: Physiologic signal archives for biomedical research, `http://www.physionet.org`
14. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. In EuroCOLT '95, pp. 23–37. Springer-Verlag: London (1995)
15. Pramanik, S., Chowdhury, U.N., Pramanik, B.K., Huda, N.: A Comparative Study of Bagging, Boosting and C4.5: The Recent Improvements in Decision Tree Learning Algorithm. Asian J. Inf. Tech. 9(6), 300–306 (2010)
16. Webb, G.I.: MultiBoosting: A Technique for Combining Boosting and Wagging. Mach. Learn. 40, 159–239 (2000)
17. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
18. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation Forest: A New Ensemble Method. Pattern Anal. Mach. Intell. 28(10), 1619–1630 (2006)
19. Kuncheva, L.I., Rodriguez, J.J.: An Experimental Study on Rotation Forest Ensembles. Lect. Notes Comput. Sci. 4472, 459–468 (2007)
20. Vapnik, V.: The nature of statistical learning theory. Springer-Verlag, New York (1995)
21. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation 13(3), 637–649 (2001)
22. Witten, I.H., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann, Burlington, MA (2011)
23. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Working set selection using second order information for training SVM. J. Mach. Learn. Res. 6, 1889–1918 (2005)
24. Chang C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Tech. 2(27), 1–27 (2011)
25. Guvenir, H.A., Acar, B., Demiroz, G., Cekin, A.: A supervised machine learning algorithm for arrhythmia analysis. In: Computers in Cardiology Conference CINC 1997, pp. 433–436. IEEE Press, New York (1997)
26. Jović, A., Bogunović, N.: HRVFrame: Java-Based Framework for Feature Extraction from Cardiac Rhythm. Lect. Notes Artif. Intell. 6747, 96–100 (2011)
27. Jović, A., Bogunović, N.: Random Forest-Based Classification of Heart Rate Variability Signals by Using Combinations of Linear and Nonlinear Features, In: Proc. of MEDICON 2010, IFMBE Vol. 29, pp. 29–32, Springer, Berlin (2010)
28. Garcia, T.B., Holtz, N.E.: 12-Lead ECG: The Art of Interpretation. Jones and Bartlett Publishers, Sudbury, MA (2001)
29. Criminisi, A., Shotton, J., Konukoglu, E.: Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Research Technical Report TR-2011-114 (2011)