



Published in final edited form as:

*Electronics (Basel)*. 2020 January ; 9(1): . doi:10.3390/electronics9010099.

## Weighted Random Forests to Improve Arrhythmia Classification

Krzysztof Gajowniczek<sup>1,\*</sup>, Iga Grzegorzczuk<sup>2</sup>, Tomasz Zbkowski<sup>1</sup>, Chandrajit Bajaj<sup>3</sup>

<sup>1</sup>Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences - SGGW, 02-776 Warsaw, Poland <sup>2</sup>Department of Physics of Complex Systems, Faculty of Physics, Warsaw University of Technology, 00-662 Warsaw, Poland <sup>3</sup>Department of Computer Science, Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78712

### Abstract

Construction of an ensemble model is a process of combining many diverse base predictive learners. It arises questions of how to weight each model and how to tune the parameters of the weighting process. The most straightforward approach is simply to average the base models. However, numerous studies have shown that a weighted ensemble can provide superior prediction results to a simple average of models. The main goals of this article are to propose a new weighting algorithm applicable for each tree in the Random Forest model and the comprehensive examination of the optimal parameter tuning. Importantly, the approach is motivated by its flexibility, good performance, stability, and resistance to overfitting. The proposed scheme is examined and evaluated on the Physionet/Computing in Cardiology Challenge 2015 data set. It consists of signals (electrocardiograms and pulsatory waveforms) from intensive care patients which triggered an alarm for five cardiac arrhythmia types (Asystole, Bradycardia, Tachycardia, Ventricular Tachycardia, and Ventricular Fultter/Fibrillation). The classification problem regards whether the alarm should or should not have been generated. It was proved that the proposed weighting approach improved classification accuracy for the three most challenging out of the five investigated arrhythmias comparing to the standard Random Forest model.

### Keywords

arrhythmia; false alarm; weighted random forest; machine learning

## 1. Introduction

Aggregation of machine learning based models is usually done by so called ensemble supervised learning [1]. The goal of ensemble algorithms is to combine the predictions of

---

Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

\* Correspondence: krzysztof\_gajowniczek@sggw.pl; Tel.: +48-506-746-850.

**Author Contributions:** K.G. prepared the simulation and analysis and wrote the Sections 1–6 of the manuscript; I.G. wrote Sections 1, 2 and 6 of the manuscript; T.Z. wrote Sections 1, 2 and 6 of the manuscript; C.B. coordinated the main theme of the research. All authors have read and agreed to the published version of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

several base models built with a given learning algorithm in order to improve robustness and generalizability over a single model [2]. There is a strong evidence, that a single model can be outperformed by an ensemble of models combined to reduce bias, variance or both [3]. A single model is unlikely to capture the entire underlying structure of the data to achieve optimal predictions. This is where integrating multiple models can significantly improve prediction accuracy. By aggregating multiple base learners (individual models), more information can be captured on the underlying structure of the data [3].

Ensembling models constitute a relevant function in data analytics and can be created in a variety of ways. The three most popular methods for combining the predictions from different models are:

- Bagging or averaging aimed at building multiple models (typically of the same type) from different subsamples of the training dataset. The driving principle is to build several estimators independently (Bagging methods [4] and Random Forests [5]) and then to average their predictions;
- Boosting aimed at building multiple models (also typically of the same type) in a sequence. Each model learns to fix the prediction errors of a prior/preceding model (e.g., AdaBoost [6] and Gradient tree Boosting [7]). Base estimators are built sequentially and in each step the last one added tries to reduce the bias of the combined estimator;
- Voting (also called stacking) aimed at building multiple models (typically of different types). Uses simple statistics (like calculating the mean) to combine predictions [4]. It is also possible to take the output of the base learners on the training data and apply another learning algorithm on them to predict the response values [8].

Each of the aforementioned methods has different characteristics. Bagging tends to reduce variance and does not work well with simple machine learning models. Boosting reduces bias by sequentially combining weak learners but is sensitive to noisy data and outliers and is susceptible to overfitting [3]. Whilst voting/stacking reduces bias by fixing the errors that base learners made by fitting one or more meta-models on the predictions made by base learners. [3,8].

As previously stated, the construction of an ensemble model is a systematic process of combining many diverse base predictive learners. When aggregating predictive learners, there is always the question of how to weight each model and how to tune the parameters of the weighting process. The most straightforward approach is simply to average the base models i.e., to give equal weight to each base model. However, numerous studies have shown that a weighted ensemble can provide superior prediction results to a simple average of models [3,9–11]. There have been several attempts in the past to improve upon the classic version of the Random Forest (RF). Attempts to improve the accuracy of classification can be broadly divided into two categories:

- Pruning of individual trees in the forest [12,13];
- Weighing individual trees [11,14–17].

Therefore, this research is focused on application of some improvements to Random Forest (RF) algorithm, especially those dedicated to development of novel weighting algorithm applicable for each tree in the Random Forest model and the comprehensive examination of the optimal parameter tuning. The tunable parameters in the proposed approach include stability of the models, error on the unseen sample, and the parameter responsible for weights distribution. Such a broad set of parameters and the approach itself are motivated by its flexibility, stability of the performance, ability to capture non-linear dependencies, and resistance to overfitting that can deliver increased predictive performance. To evaluate the proposed weighting approach, numerical experiments on five data sets regarding arrhythmia classification have been conducted. To demonstrate the generalizability of the proposed scheme we compare achieved results to the results obtained in our previous study (standard version of Random Forest) [18] examining data provided by organizers of the Physionet/ Computing in Cardiology Challenge [19]. During the challenge, participants were provided with 750 signals registered 5 min before the alarm generation and information about what type of arrhythmia caused the alarm. The alarms were triggered by five types of arrhythmia: Asystole, Bradycardia, Tachycardia, Ventricular Fibrillation or Flutter, and Ventricular Tachycardia. All the signals were analyzed by expert annotators and labelled as true or false [18,19].

The main questions that we want to address in this paper are:

- Does the proposed weighting method introduce improvements to the standard Random Forest algorithm?
- To what extent is it possible to outperform previous results of reducing false arrhythmia alarms?
- What is the effect of different tuning parameters as part of finding optimized ensembles on the quality of predictions?
- Can the results be generalized over different arrhythmia types (datasets of different characteristics)?

The remainder of this paper is organized as follows: Section 2 provides an overview of the similar research problems for predictive models aggregation, weighting schemas, and arrhythmia classification. In Section 3, the theoretical framework of the weighted Random Forest algorithm is presented. In Section 4, the research framework was outlined, including the details of numerical implementation, feature vector description and model performance measures. Section 5 outlines the experiments and presents the discussion of the results. The paper ends with concluding remarks in Section 6.

## 2. Literature Review

There have been extensive studies on weighted ensembles in the literature. Shahhosseini et al. [3] propose a systematic approach to find the optimal weights to create ensembles for bias-variance tradeoff using cross-validation for regression problems (Cross-validated Optimal Weighted Ensemble (COWE)). It is known, that tuning hyperparameters of each base learner during the ensemble weight optimization process can produce better performing

ensembles. Therefore, authors proposed a nested algorithm based on bi-level optimization that considers tuning hyperparameters as well as finding the optimal weights to combine ensembles (Cross-validated Optimal Weighted Ensemble with Internally Tuned Hyperparameters (COWE-ITH)). Later on, Pham et al. [14] proposed a weighted scheme that generalizes bagged ensemble learning to a weighted vote by considering different ways of averaging. Kuncheva et al. [20] proposed a probabilistic framework for classifier combination. It gives rigorous optimality conditions (minimum classification error) for four combination methods: majority vote, weighted majority vote, recall combiner, and the naive Bayes combiner. The framework is based on two assumptions: class-conditional independence of the classifier outputs and an assumption about the individual accuracies. Filmus et al. [21] analyzed the weighted voting games (WVGs) i.e., a class of cooperative games that capture settings of group decision making in various domains, such as parliaments or committees.

In Breiman's [5] forests, the final prediction is the average of the individual tree outcomes. A natural way to improve the method is to incorporate tree-level weights to emphasize more accurate trees in prediction. The concept of weighted trees in the Random Forest is not entirely new. Winham et al. [11] described the weighted Random Forest method, which incorporates tree-level weights into the usual RF algorithm to emphasize more accurate trees in prediction and calculation of variable importance. They considered different tree-weights, and present simulations to compare the performance of the proposed approach to the traditional RF algorithm both in terms of prediction accuracy and performance of variable importance measures. Pham et al. [22] described a potential improvement on the Random Forest (for the binary classification problem) using Cesaro average. This method is motivated by the potential instability of averaging predictions of trees. Byeon et al. [15] used Out-of-Bag (OOB) samples for deriving Akaike weights while averaging the tree results. Based on the proposed algorithm they analyzed in-depth the consumers' demand level in order to operate "the Voucher Program for Speech Language Therapy" efficiently and for suggesting different ways to improve the special education service support. Xuan et al. [16] introduced Refined Weighted Random Forests (RWRF) to credit card fraud detection. The improvement regards two aspects. They used all training data (including In-Bag (INB) data and OOB data) because performance evaluation of different base classifiers should have the same evaluation dataset. Moreover, they used margin between probability of predicting true class and false class label which measures the extent to which the expected number of votes for the right class exceeds the expected number of votes for other class. Kulkarni et al. [17] presented attempts to improve performance of Random Forest classifier in terms of accuracy, and time required for learning and classification. They are based on disjoint partitions of training datasets, use of different attribute evaluation/split measures to induce base decision trees of Random Forest, application of weighted voting instead of majority voting, use of diversity among bootstrap datasets to generate maximum diverse classifiers, and application of dynamic programming approach to find optimal subset of Random Forest.

Moreover, there are multiple, different application areas in which weighted ensemble approaches (especially Random Forests) are used. Booth et al. [23] introduced an automated trading system based on performance weighted ensembles of Random Forests that improves the profitability and stability of trading seasonality events. The results show that recency-

weighted ensembles of Random Forests produce superior results, in terms of both profitability and prediction accuracy, compared to other ensemble techniques. Utkin et al. [24] proposed weighted Random Survival Forest which can be regarded as a modification of the Random Forest improving its performance. The main idea underlying the proposed model is to replace the standard procedure of averaging used for estimation of the Random Survival Forest hazard function by weighted averaging. The weights are assigned to every tree and can be viewed as training parameters which are computed by solving a standard quadratic optimization problem maximizing Harrell's C-index. Finally, Sunil Babu et al. [25] utilized an effective meta-heuristic feature selection technique along with hybrid Naive Bayes (NB) and sample weighted Random Forest (SWRF) classification approach for sub-acute ischemic stroke lesion segmentation. Here the NB classifier is trained and applied to estimate the weights of training samples. Then, the training samples with estimated weights are utilized to train SWRF. Brief comparison of the weighting methods applied for ensemble classifiers is presented in Table 1.

Automatic detection and classification of life-threatening arrhythmia plays an important role in dealing with various cardiac conditions. An arrhythmia occurs when the heartbeat is irregular. Some arrhythmias are defined only by the frequency of heart contractions. Bradycardia occurs when the number of heartbeats is less than 40 per minute (bpm). Tachycardia is diagnosed when heart rate exceeds 140 beats per minute. The Asystole occurs where there are no heart contractions detected for at least four seconds. Ventricular Flutter or Fibrillation and Ventricular Tachycardia are arrhythmias that not only have an abnormally fast rhythm, but the morphology of QRS complexes differs from normal. Such precise definitions make arrhythmias easy to detect from an algorithmic point of view, but only when high quality data is available. In real life, signals are often noisy and contain artifacts, which is why it is so important to use robust algorithms to locate the heartbeats within the signal.

To detect and to analyze Bradycardia, Tachycardia, and Asystole the key is to correctly identify locations of QRS complexes (or in terms of Asystole, lack of them). The electrocardiography (ECG) signal is the main source of information about the patient's cardiovascular condition, but because the measurements are taken from the patient's body motion artifacts, sweating or muscle spasms can impact the signal. Therefore, particular attention should be paid to assessing the quality of the data. In bedside monitors, in addition to ECG, pulsatile waveforms such as blood pressure (BP) or plethysmogram (PLETH) are also recorded. Such recordings may provide necessary information about the heart rate when ECG signal quality is very low. In the first row of Table 2, we present methods commonly used for detection of QRS complexes. First, are methods based on ECG signal analysis only and then algorithms using supplementary information gathered from other medical signals (e.g., blood pressure) as well. Next, there is a section dedicated to each type of arrhythmia and the methods of diagnosing it (if applicable). The last row of the table contains more general methods used to classify all types of arrhythmias. They were used with a different parameter configuration for each arrhythmia.

### 3. Weighted Random Forest

Let's consider a binary classification problem, having a training sample ( $\mathbf{D}$ ) of  $n$  ( $i \in \{1, \dots, n\}$ ) observations, a class variable  $Y_i \in \{0, 1\}$ , and  $p$  predictor variables  $X_1, \dots, X_p$  ( $k \in \{1, \dots, p\}$ ). The main objective is to find a model for predicting the values of  $Y_i$  from new  $\mathbf{X}$  values [43].

Random Forest algorithm incorporates the bagging procedure to each decision tree (base learner) [33], generating  $Ntree$  ( $j \in \{1, \dots, Ntree\}$ ) new training datasets  $\mathbf{D}_j$  (also called In-bag sample; further denoted as  $INB$ ), each of size  $n'$  by sampling from  $\mathbf{D}$  uniformly and with replacement (this causes that some observations might be duplicated). Each time In-bag sample is expected to have approximately  $1 - \frac{1}{e} \approx 63.2\%$  of the unique observations, while rest of them fall into the so called Out-of-Bag sample (further denoted as  $OOB$ ). The RF algorithm has an additional sampling stage (called Random Decision Tree) selecting a random subset of the features  $mtry$  at each candidate split in the learning process. Typically, for a classification problem with  $p$  features,  $\text{floor}(\sqrt{p})$  features are used in each split [52].

The original implementation of Random Forest aggregates tree-level results equally across trees obtaining the final probability  $\hat{Y}_i^{RF}$  (so called score) for a given observation using one of the following formulas:

$$\hat{Y}_i^{RF} = \sum_{j=1}^{Ntree} I[\hat{Y}_{ij} > 0.5], \quad (1)$$

where  $I$  is the indicator function,  $\hat{Y}_{ij}$  is the probability for  $i$ -th observation assigned by the  $j$ -th tree. The above formula uses majority voting with predefined threshold set at 0.5, which sometimes might not be an appropriate approach. Therefore, to gain some flexibility (and to be more sensitive for any possible deviations from the desired output), in this article the following formula is used:

$$\hat{Y}_i^{RF} = \frac{1}{Ntree} \sum_{j=1}^{Ntree} \hat{Y}_{ij}, \quad (2)$$

which is in other words a simple average of the probabilities from each tree.

In this research, we implemented the usual RF algorithm to build the trees of the forest, however, we utilize performance-based weights for tree aggregation. In particular, we considered weighting probability from each tree in the forest, such that better performing trees are weighted more heavily:

$$\hat{Y}_i^{RF} = \frac{1}{Ntree} \sum_{j=1}^{Ntree} \hat{Y}_{ij} * w_j. \quad (3)$$

Because weights are based on the performance of the particular/a distinct tree, applying the weights to the same dataset from which they were calculated (as was done in [38,53]) would bias prediction error assessment. Therefore, estimates of the predictive ability of each tree are calculated using both the *INB* and the *OOB* observations, which can be further used to calculate weights. In order to obtain trees having good generalization abilities, we introduced a special tree performance measure ( $\theta$ ) incorporating weighted Area Under the Curve (AUC). Trees having the greatest values of the following measure should have higher weights (in Equation (3)) while determining the final probability:

$$\theta = f(AUC_{INB}, AUC_{OOB}) = -\alpha |AUC(w^{obs})_{INB} - AUC_{OOB}| + (1 - \alpha) AUC(w^{obs})_{OOB} \quad (4)$$

where  $AUC_{INB}$  stands for the In-Bag sample accuracy,  $AUC_{OOB}$  for the Out-of-Bag sample accuracy and  $w^{obs}$  is an observation level weight vector (please see next paragraph). Parameter  $\alpha$  measures the weights of the first and the second term in the equation i.e., it controls what is more important during learning, stability of the tree, or small errors on the unseen dataset [54].

It should be noticed that some observations are more difficult to correctly classify than others. That is why some algorithms incorporate observation weighting. A classic example is the AdaBoost algorithm (learning in iterative/sequential manner) where for each iteration the observation weights are individually modified and the classification algorithm is reapplied to the weighted observations. At each step, those observations that were misclassified by the model induced at the previous step have their weights increased, whereas the weights for those classified correctly are decreased. Thus, as iterations proceed, observations that are difficult to be classified correctly receive ever-increasing influence. Each successive model is thereby forced to concentrate on those training observations that are missed by previous ones in the sequence [55].

Observation weighting can be also considered as a fairness problem [56]. By definition, fairness is the absence of any prejudice or favoritism towards an individual or a group based on their intrinsic or acquired traits in the context of decision-making [56]. Thus, an unfair algorithm is one whose decisions are skewed toward a particular group. For instance, in Random Forest some trees might have better performance on an OOB sample because they were trained on the similar observations (INB sample). Such observations are hard to predict correctly by the trees that did not have them in INB used for training.

In this article, we used this idea for estimating the performance of each tree. Weighted AUC can handle such weights implicitly so instead of using 1 for a given example, it uses the specified weight  $w_i$  derived based on the observations in Out-of-Bag sample:

$$w_i^{obs} = \frac{1}{\# OOB} \sum_{j \in OOB} |r_{ij} - \hat{r}_{ij}|. \quad (5)$$

If for instance  $w_i^{obs} = 0$  (which means that each tree correctly predicted the class) the example is practically ignored. As a result, the miss-classified examples have more influence on the final performance of a given tree.

In order to estimate weighted AUC (Formula (4)), as we assumed  $\hat{Y}_i \in \mathbb{R}$  is the predicted score for each observation, let  $\Gamma_1 = \{i: Y_i = 1\}$  be the set of positive examples and  $\Gamma_0 = \{i: Y_i = 0\}$  be the set of negative examples. Then, the total positive weight is  $W_1 = \sum_{i \in \Gamma_1} w_i^{obs}$  and the total negative weight is  $W_0 = \sum_{i \in \Gamma_0} w_i^{obs}$  [57]. Moreover, for any threshold  $\tau \in \mathbb{R}$ , we defined the thresholding function  $t_\tau: \mathbb{R} \rightarrow \{0, 1\}$  such as:

$$t_\tau(\hat{Y}) = \begin{cases} 1, & \hat{Y} \geq \tau \\ 0, & \hat{Y} < \tau \end{cases} \quad (6)$$

Based on the above formula the weighted false positive rate is defined as:

$$FPR(\tau) = \frac{1}{W_0} \sum_{i \in \Gamma_0} I[t_\tau(\hat{Y}_i) \neq 0] w_i^{obs}, \quad (7)$$

where  $I$  is the indicator function that is 0 for a correct prediction, and 1 otherwise. On the other hand, the weighted the true positive rate is defined as:

$$TPR(\tau) = \frac{1}{W_1} \sum_{i \in \Gamma_1} I[t_\tau(\hat{Y}_i) = 1] w_i^{obs}. \quad (8)$$

A weighted ROC curve is drawn by plotting  $FPR(\tau)$  and  $TPR(\tau)$  for all thresholds  $\tau \in \mathbb{R}$ . Finally, the weighted AUC ( $AUC(w^{obs})$ ) measure can be calculated using the trapezoid rule (integral), which is known to be extremely accurate when approximating the definite integral of periodic functions [57].

Next, a question arises: how much greater should be weights for the best trees? In the literature, there are several approaches proposed to determine weights in multi-criteria decision making [58]. One of them incorporates ranked weights. The rank order weight determination is comprised of two stages:

- Ranking the pre-defined criterion ( $\Theta$ ) according to their importance (performance of the tree derived using Formula (4));
- Weighting the criteria from their ranks using some rank order weighting approach.

In other words, let's assume having a list of  $Ntree$  prioritized (ranked) criteria, where each criterion  $j$  has a rank  $r_j$  ( $j = 1, \dots, Ntree$ ). The goal is to select and rank a set of  $Ntree$  criteria that seems to be relevant, giving each  $j$ -th criterion a rank  $r_j$ . The rank is inversely related to weight, which means that first rank  $r_1 = 1$  denotes the highest weight (best tree), whilst rank



$r_{Ntree} = Ntree$  denotes the lowest weight (worst tree). Many authors suggest various approaches for assigning weights based on a given criterion e.g., rank reciprocal (inverse), rank sum (linear), and rank exponent weights [59]. In this paper, we assumed that weights should be exponential [58]:

$$w_j = \frac{(Ntree - r_j + 1)^p}{\sum_{k=1}^{Ntree} (Ntree - r_k + 1)^p} \quad (9)$$

Where  $r_j$  is the rank of the  $j$ -th tree,  $p$  is the exponential parameter describing the strength of the weights. All weights are normalized and, in consequence, sum up to 1. An example of weights' influence on the final probability (Equation (3)) inducted by the parameter  $p$  is presented in Table 3 and Figure 1 below.

In this example, the best performance in terms of Equation (4) was the third tree (fourth column). Because parameter  $p$  is set to 2 the nominator of the exponential rank weight equals 16 ( $4 - 1 + 1$ )<sup>2</sup> and after the normalization (by denominator) 0.533. Analyzing Figure 1, it can be stated that when parameter  $p$  equals 0 it is standard aggregation, where each tree has the same weight. Whilst  $p = 1$  represents linear rank sum weight. When  $p$  increases, the weights distribution becomes steeper, i.e., the greater the parameter  $p$  is the final probability is more influenced by the trees having better performance.

Like any weighting scheme, determining where to apply a particular weight is of utmost importance. With a Random Forest with  $j$  (usually 500) trees,  $mtry$  features used in each split (here  $\text{floor}(\sqrt{p})$ ) and  $D$  training dataset the complete Weighted Random Forest algorithm pseudocode is summarized in Algorithm 1 (below).

#### Algorithm 1.

Weighted Random Forest algorithm pseudocode.

---

**input:** Number of Trees ( $Ntree$ ), random subset of the features ( $mtry$ ), training dataset ( $D$ )

**output:** Random Forest ( $RF$ )

1:  $RF$  is empty

2: **for each**  $j$  to  $Ntree$  **do**

3:  $D_j =$  Bootstrap Sample ( $D$ )

4:  $DT_j =$  Random Decision Tree ( $D_j$ ,  $mtry$ )

5:  $RF = RF \cup DT_j$

6: **end**

7: **for each**  $i$  to  $n$  **do**

8: Compute  $w_i^{obs}$  using Formula (5)

9: **end**

10: **for each**  $j$  to  $Ntree$  **do**

11:  $\theta_j = f\left(AUC(w^{obs})_{INB_j}, AUC(w^{obs})_{OOB_j}\right)$

```

12: end
13: for each  $j$  to  $N_{tree}$  do
14: Compute  $w_j$  using Formula (9)
15: end
16: for each  $i$  to  $n$  do
17: Compute final prediction  $\hat{Y}_i^{RF}$  using Formula (3)
18: end
19: return  $RF$ 

```

---

Our implementation may suggest that this algorithm is applicable to Random Forest only. However, it can be generalized to any ensemble consisting of any kind of  $j$  base models. Which is done by changing observations weighting based on the Out-of-Bag samples (Formula (5), line 8 in the pseudo-code above) by any other sample e.g., training dataset.

## 4. Research Framework and Settings

### 4.1. Feature Vector

Database used in this paper consisted of 750 multi-signal recordings registered for patients on Intensive Care Unit (ICU). Registered signals were 5 min in length (sampling frequency 250Hz) and ended with an alarm generated for one of five types of arrhythmia. Each of the recordings had two leads of the ECG, at least one pulsatile waveform (arterial blood pressure (ABP) or photoplethysmogram (PLETH)) and respiratory signal. Distribution of the recordings among the investigated five types of arrhythmias and whether the alarm should or should not have been generated are: Asystole (No—100, Yes—22); Bradycardia (No—43, Yes—46); Tachycardia (No—9, Yes—131); Ventricular Tachycardia (No—252, Yes—89); Ventricular Fibrillation or Flutter (No—52, Yes—6). The signals provided were already pre-filtered with multiple notch filters and finite impulse response (FIR) band pass filter (0.05–40 Hz) [18,19].

As mentioned in Section 2, to diagnose Asystole, Bradycardia, and Tachycardia it is critical to properly locate consecutive heart beats. Hence, the first step to create features was the detection of QRS complexes in the ECG signal, performed by a low-complexity R-peak detector as described in [18,51]. At the same time, an open source *wabp* algorithm [19] was used to locate the beats in pulsatile waveforms provided (ABP, PLETH). The quality of the beats detection was assessed by comparing obtained QRS locations among the signals. Each located beat was marked as true positive (TP)—if it was found in both signals or false positive (FP) and negative (FN), respectively, in compared signals, depending on the order in which they were being compared [18,51]. Then, an F1-score as  $F1 = 2 * TP / (2 * TP + FP + FN)$  was calculated. The more beat locations were similar in the signals, the closer F1-score was to 1. If there were no matches of beat locations the F1-score equaled 0. The two signals with the highest F1-score were considered in the following analysis [18,51].

In diagnosing Ventricular Flutter or Fibrillation and Ventricular Tachycardia, the features were generated with spectral purity index (SPI) [44,51]. The reason why, these arrhythmias

require a different method of detection of physiological QRS complexes as it is impossible due to the nature of ventricular originated arrhythmias (see Section 2). The maxima and minima of obtained SPIs were used as features to check if the alarm should have occurred i.e., max and min SPI for Ventricular Tachycardia and max SPI for Ventricular Fibrillation or Flutter.

#### 4.2. Numerical Implementation

All numerical experiments presented below were prepared using *R* programming language [60] working on Ubuntu 18.04 operating system on a personal computer equipped with Intel Core i7-9750H 2.6 GHz processor (12 threads) and 32 GB RAM. The entire weighted Random Forest algorithm was built based on own modification of the following R libraries: *ranger*—implementing in *C++* and *R* the state-of-the-art Breiman's RF, especially suited for high dimensional data [61]; *pROC*—for finding optimal threshold for class determining based on the Youden Index [62]; *WeightedROC*—for fast computation of Receiver Operating Characteristic curves and Area Under the Curve for weighted binary classification problems. Furthermore, to perform this study, many wrapper functions working on the output from the *ranger* package were written, e.g., functions extracting the class probability matrix or In-Bag/Out-of-Bag matrix for each observation and tree [18].

The estimates for the performance measures for the training and validation samples were produced with *k*-fold cross-validation [18]. The number of *k* sets was set to 10 when there were more than 10 samples in the smaller class. Otherwise, *k* was set to the size of the smaller class to ensure that there was at least one sample from both of the classes (e.g., six for ventricular fibrillation or flutter) [18]. The *k* sets were generated so that the class distribution in every set represented the class distribution of the entire dataset using stratified sampling based on the *createFolds* function implemented in the *caret* library [63]. All further results are presented as an average over *k*-folds.

#### 4.3. Performance Measures

A proper evaluation is crucial for models built with any statistical learning algorithm. Hence, in this research different types of evaluation metrics were used. The main measure used for this purpose in the Challenge is the Score measure defined as:

$$\text{Score} = \frac{100 \times (TP + TN)}{TP + FP + TN + 5 \times FN'} \quad (10)$$

where for a binary classification problem the following indications are used [64]: *TP* and *TN* denote number of correctly classified either positive or negative instances, *FP* stands for the instances predicted as Yes when the actual output was No, and finally *FN* indicate the number of instances predicted as No while the actual output was Yes. According to the above formula it can be seen that the measure was designed to treat *FN*-false negative (genuinely life-threatening events that the program considered unimportant) especially harshly [18,51].

The second used measure is Area Under the ROC Curve, which is particularly important in this research since it was used to tune the parameters of each model [18]. The construction

of the ROC curve and the calculation of the AUC measure was described in Section 3 [52,54].

#### 4.4. Benchmarking Methods

In order to compare and assess the quality of the proposed weighting approach, we treated results obtained in our previous study as a benchmark [18] i.e., the standard Random Forest, where final probability (requires AUC calculation) for each observation is derived using majority class voting with the cutoff set to 0.5 (Voting Prob 0.5) and the final class for each observation incorporates a cutoff set to 0.05 (Vote 0.5 0.5).

In addition to the above method, the following benchmarking algorithms have been used. First is the CART algorithm implementing classification and regression trees (*rpart* library) [65]. It utilizes pruning during the growth stage. In order to generalize the knowledge, this approach prevents new splits from being created when the previous splits provided only a slight increase in predictive accuracy. The complexity parameter  $cp$  varied from 0 to 0.1 in increments of 0.01.

The second model has been built based on the framework described in [66], in order to construct a model of the support vector machine, the C-SVM function from the *kernelab* library was used. The linear, polynomial (degrees 1, 2, and 3) and radial ( $\gamma$  from 0.1 to 1, by 0.2) kernel functions were used and  $\epsilon$  (which determines the margin width for which the error function is zero) was arbitrarily taken from the following set of {0.1, 0.3, 0.5, 0.7, 0.9}. The regularized parameter  $C$  that controls overfitting has been arbitrarily set to one of the following values {0, 0.2, 0.4, 0.6, 0.8, 1}.

The last results were derived based on the AdaBoost algorithm implemented in the *adabag* library. In this case, the number of iterations for which boosting is run (or the number of trees to use) has been set to 100 iterations, the weight updating coefficient  $\alpha$  has been set to  $1/2 \ln((1 - error)/error)$ , and finally each tree was at most 5 levels deep.

Finally, in each tuning parameter process, a structure maximizing the Score function (Equation (10)) on the training dataset has been chosen as a final structure of the model.

#### 4.5. Tuning of the Weighting Parameters

To answer the question about the effect of the tuning of the weighting parameters in terms of the quality of predictions we performed a grid search checking various combinations of the  $\alpha$  and  $p$  parameters:

To check the effect of the weighting parameters tuning, in terms of predictions quality, we performed a grid search checking various combinations of  $\alpha$  and  $p$  parameters:

- $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ —controlling the importance of the first (model stability) or the second (small error on the unseen dataset) term in Formula (4).
- $p \in \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ —which is the exponential parameter describing the strength of the weights (distribution).

It would purely be an explanatory analysis, giving an insight into investigated phenomena.

## 5. Empirical Analysis

The classification performance of the proposed approach was evaluated with AUC and a Challenge Score (Formula (10)) within different datasets (In-Bag, Out-of-Bag, Training, and Validation). In the tables below, we present exemplary results, as values obtained for Validation dataset (Tables 4–8). This dataset was chosen as representative, since the Validation sample did not take part during the model training. The results were calculated as average values based on  $k$ -cross validation.

Each table first presents results for AUC measure and then for the Score obtained using standard Random Forest algorithm (column Base). The next column includes different values of the parameter  $\alpha$  and the headings of the subsequent columns show values of the parameter  $p$ . Finally, intersections of consecutive values of  $\alpha$  and  $p$  parameters present/demonstrate improvement (or deterioration) for all aforementioned measures. It should be noted that  $p = 0$  leads to equal weights (as in case of standard Random Forest). For quick identification of desired/undesired combinations we set a proper color scale i.e., improvements are marked in (light) green while deterioration is marked in (light) orange.

In Table 4, the AUC for Asystole based on the standard Random Forest is 0.93. Applying weighted version results with an improvement of 0.055 from most of the combinations gives an AUC of 0.985. In the bottom-right corner we can see an even greater improvement of 0.060. For this type of arrhythmia, Score values show constant behavior with an improvement of 19.00 units ( $61.67 + 19 = 80.75$ ). For Bradycardia, we can observe diverse results for both AUC and Score (Table 5). The left-bottom corner for the AUC presents intersections where applying weights can improve results (approximately 0.001). Other regions of the table show a slight performance deterioration for both measures.

Results for Tachycardia (Table 6), in terms of AUC, present either infinitesimal improvement, for all ranges of  $\alpha$  when the exponential parameter is less than about 2.5 or slight deterioration of approximately 0.008 while increasing the alpha to above 2.5. Score is constant except for  $p = 5$  and  $\alpha = 0.6$ .

The AUC for the base RF model for Ventricular Fibrillation or Flutter reaches 0.97 (Table 7). Applying weighted Random Forest resulted in almost ideal classification, since the measure is above 99.5 for all combinations of the tuned parameters. Such improvement influences also Score measure which increases of 55.55 for most intersections (some combinations boosted results of 54.51). Results for Ventricular Tachycardia (Table 8) shows that the best results for the Score can be obtained when weights vanish linearly ( $p$  equals 1) for alpha parameters less than 0.9. Slightly worse results can be obtained while using  $\alpha$  between 0.7 and 1 and  $p$  between 2.5 and 4.5 (middle-bottom part of the Table 8).

As presented in Table 9, there are three types of arrhythmia i.e., Asystole, Ventricular Tachycardia, and Ventricular Fibrillation or Flutter which are difficult to predict, when looking into the results for the validation dataset. However, application of weighted RF resulted in improved classification. Importantly, the performance of weighted RF models in

terms of the Score measure observed on the validation sample confirms that the method works well and is able to capture arrhythmias with high accuracy. The following Scores were obtained for the proposed approach:

- 31.9 for Ventricular Tachycardia—accuracy of the model was improved in comparison to RPART (27.7), C-SVM (30.5), and AdaBoost (29.9);
- 86.1 for Ventricular Fibrillation or Flutter—accuracy of the model was improved in comparison to RPART (29.4), C-SVM (50.1), and AdaBoost (50.1);
- 80.7 for Asystole—accuracy of the model was improved in comparison to RPART (52.1), C-SVM (61.7), and AdaBoost (61.7).

Finally, in order to assess whether differences in Score measures for all methods presented in Table 9 are statistically significant, we have used Nemenyi post-hoc tests [67] (test is performed by taking into account each of the  $k$ -folds). This is a post-hoc test intended to find the groups of data that differ after a statistical test of multiple comparisons (such as the Friedman test) has rejected the null hypothesis that the performance of the comparisons on the groups of data is similar. The test makes pair-wise tests of performance. Moreover, for a better understanding of the results the Critical Difference (CD) diagram depicted in Figure 2 has been created. This diagram provides an interesting visualization of the statistical significance of the observed paired differences between a set of models on a set of different datasets (i.e., dataset for each arrhythmia type). It allows one to compare all of the models against each other on these datasets and check the results of all these paired comparisons.

In the resulting diagram, each model is represented in rows (with an average rank position across  $k$ -folds). The null hypothesis is that the average ranks of each pair of models do not differ with statistical significance (at confidence level  $\alpha = 0.95$ ). Vertical lines connect the models for which we cannot exclude the hypothesis that their average ranks are equal. Any pair of models whose lines are not connected can be seen as having an average rank that is different with statistical significance. On top of the graph, the critical difference depicts the required difference between the average ranks for the two pair of models to be considered significantly different.

Based on the Nemenyi diagram, it can be stated that improvement in results for Asystole are statistically significant in comparison to all the benchmarking methods. For Ventricular Tachycardia, despite the slight improvement in this measure (see Table 9), the difference between standard and weighted RF is insignificant. On the other hand, Random Forests give statistically significant better results than other methods. Comparing standard and weighted RF it can be seen that results improvement is statistically significant for Ventricular Fibrillation or Flutter.

The results indicate that weighting applied to Random Forests can be considered a viable approach aimed at improvement of the classification especially in case when the benchmarking methods like decision trees, while support vector machines are not able to deliver acceptable accuracy.

## 6. Conclusions

Over recent years, the idea to aggregate the machine learning; based models has been extensively studied and documented in the literature. Researchers have often shown that combining the predictions of several base models, built with a given learning; algorithm, improves robustness and generalizability over a single model. Ensemble models such as Bagging;, Boosting, Random Forests, and stacking have different characteristics. Some approaches do not work well with relatively simple models or are sensitive to noisy data and outliers.

While aggregating predictions from the base models, there is always a question whether better models should have greater influence on the final performance. The easiest way is to treat all predictions equally while more complex approaches assign bigger weights to superior models. The comprehensive ensemble framework should simultaneously tune various parameters (including stability of the models), error on the unseen sample and the parameter responsible for weight distributions. Therefore, the motivation of this study was to propose the new weighting algorithm applicable for standard Random Forest algorithm and the comprehensive examination of the optimal parameter tuning. To demonstrate the generalizability of the proposed scheme we conducted the study based on the Physionet/ Computing in Cardiology Challenge, 2015. The Challenge addresses the issue of false arrhythmia alarms in the Intensive Care Units, which has detrimental consequences for both patients and medical staff. The analyzed dataset consists of medical signals (ECG, BP, and PLETH) for which triggered arrhythmia al arms and the classification problem is to solve whether the generated alarm was true of false. It is a challenging task for any classification algorithm as there could be a number of triggers for false alarms, e.g., the noises and machine malfunctions affecting the; signals and in consequence highly influencing the model performance.

The novelty of the research was demonstrated through weighting algorithm design to enhance the Random Forest model. It proves that the proposed solution is robust and can deal not only with easily diagnosable arrhythmias like Asystole or Bradycardia but also with arrhythmia types which are difficult to predict, i.e., Ventricular Tachycardia and Ventricular Flutter and Fibrillation. The results proved that the weighted Random Forest is able to detect three arrhythmias (i.e., Ventricular Tachycardia, Ventricular Flutter and Fibrillation, and Asystole) with better accuracy than the benchmarking standard Random Forest. In the case of Bradycardia and Tachycardia, after proper tuning of the parameters, the classification accuracy can be slightly improved too. Although, it is important to acknowledge, that the accuracy of Bradycardia and Tachycardia detection is already high, as these are arrhythmias without any morphological changes in the ECG signal, which makes them relatively easy to predict. Finally, comparing our proposed algorithm with three state-of-the-art benchmarking methods one can see that for Asystole and Ventricular Tachycardia results improvements are statistically significant.

The authors believe that the problem to reduce the number of false alarms, while avoiding the suppression of true ones, is valid and therefore, the study can be extended further with

the application of other other weighting approaches to obtain higher classification accuracy. This may lead to further study on algorithms' diversity and their effects on performance.

Future work in this area should include extending this weighted ensemble framework to multiclass classification and a regression problem. Secondly, applying a similar concept on other ensemble creating methods. We intend to further explore the performance and default parameter settings in the context of the bias and variance of the base classifier, with potentially both a theoretical and empirical analysis.

## Acknowledgments

**Funding:** This research received no external funding.

## References

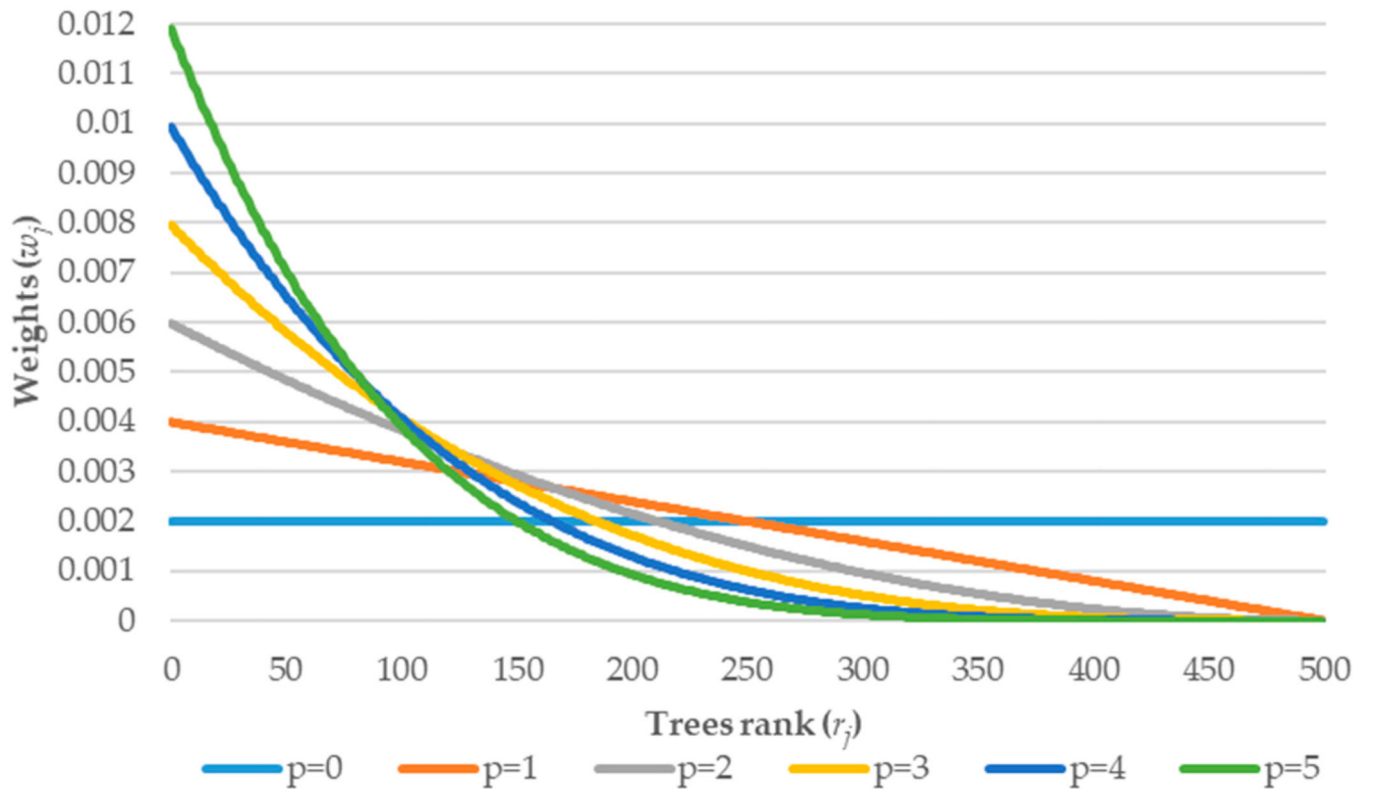
1. Sagi O; Rokach L Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov* 2018, 8, e1249.
2. Ren Y; Zhang L; Suganthan PN Ensemble Classification and Regression-Recent Developments, Applications and Future Directions. *IEEE Comput. Intell. Mag* 2016, 11, 41–53.
3. Shahhosseini M; Hu G; Pham H Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems. *arXiv* 2019, arXiv:1908.05287
4. Breiman L Bagging predictors. *Mach. Learn* 1996, 24, 123–140.
5. Breiman L Random forests. *Mach. Learn* 2001, 45, 261–277.
6. Freund Y; Schapire RE Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996*; pp. 148–156.
7. Friedman JH Stochastic gradient boosting. *Comput. Stat. Data Anal* 2002, 38, 367–378.
8. Large J; Lines J; Bagnall A A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. *Data Min. Knowl. Discov* 2019, 33, 1674–1709. [PubMed: 31632184]
9. Bhasuran B; Murugesan G; Abdulkadhar S; Natarajan J Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases. *J. Biomed. Inform* 2016, 64, 1–9. [PubMed: 27634494]
10. Ekbal A; Saha S Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowl.-Based Syst* 2013, 46, 22–32.
11. Winham SJ; Freimuth RR; Biernacka JM A weighted random forests approach to improve predictive performance. *Stat. Anal. Data Min. ASA Data Sci. J* 2013, 6, 496–505.
12. Martínez-Muñoz G; Suárez A Using boosting to prune bagging ensembles. *Pattern Recognit. Lett* 2007, 28, 156–165.
13. Wang M; Zhang H Search for the smallest random forest. *Stat. Interface* 2009, 2, 381–388. [PubMed: 20165560]
14. Pham H; Olafsson S Bagged ensembles with tunable parameters. *Comput. Intell* 2019, 35, 184–203.
15. Byeon H; Cha S; Lim K Exploring Factors Associated with Voucher Program for Speech Language Therapy for the Preschoolers of Parents with Communication Disorder using Weighted Random Forests. *Int. J. Adv. Comput. Sci. Appl* 2019, 10.
16. Xuan S; Liu G; Li Z Refined Weighted Random Forest and Its Application to Credit Card Fraud Detection. *Lect. Notes Comput. Sci* 2018, 11280, 343–355.
17. Kulkarni VY; Sinha PK Effective learning and classification using random forest algorithm. *Int. J. Eng. Innov. Technol* 2014, 3, 267–273.
18. Gajowniczek K; Grzegorzczak I; Zbkowski T Reducing False Arrhythmia Alarms Using Different Methods of Probability and Class Assignment in Random Forest Learning Methods. *Sensors* 2019, 19, 1588.



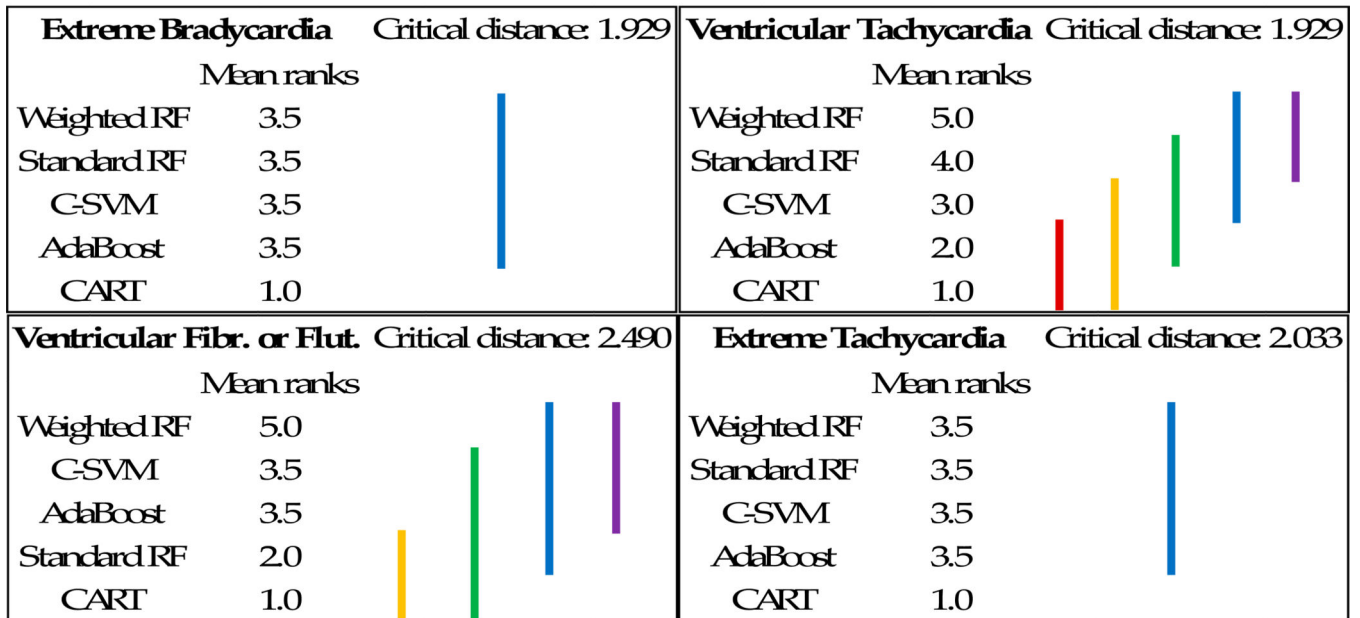
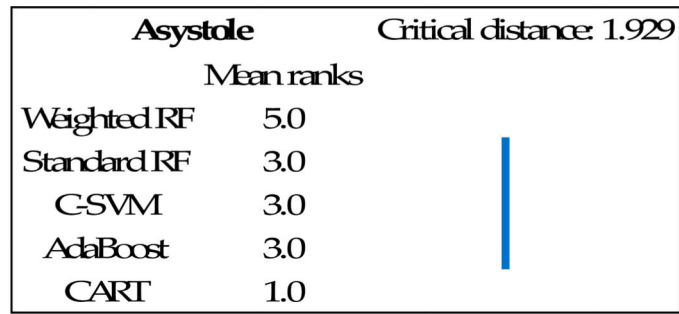
19. Clifford GD; Silva I; Moody B; Li Q; Kella D; Shahin A; Kooistra TL; Perry D; Mark RG The PhysioNet/computing in cardiology challenge 2015: Reducing false arrhythmia alarms in the ICU. In Proceedings of the 2015 Computing in Cardiology Conference (CinC), Nice, France, 6–9 September 2015; pp. 273–276.
20. Kuncheva LI; Rodríguez JJ A weighted voting framework for classifiers ensembles. *Knowl. Inf. Syst* 2012, 38, 259–275.
21. Filmus Y; Oren J; Zick Y; Bachrach Y Analyzing Power in Weighted Voting Games with Super-Increasing Weights. *Theory Comput. Syst* 2018, 63, 150–174.
22. Pham H; Olafsson S On Cesaro averages for weighted trees in the random forest. *J. Classif* 2019, 1–14.
23. Booth A; Gerding E; McGroarty F Automated trading with performance weighted random forests and seasonality. *Expert Syst. Appl* 2014, 41, 3651–3661.
24. Utkin LV; Konstantinov AV; Chukanov VS; Kots MV; Ryabinin MA; Meldo AA A weighted random survival forest. *Knowl.-Based Syst* 2019, 177, 136–144.
25. SunilBabu M; Vijayalakshmi V An Effective Approach for Sub-acute Ischemic Stroke Lesion Segmentation by Adopting Meta-Heuristics Feature Selection Technique Along with Hybrid Naive Bayes and Sample-Weighted Random Forest Classification. *Sens. Imaging* 2019, 20.
26. Pan J; Tompkins WJ A Real-Time QRS Detection Algorithm. *IEEE Trans. Biomed. Eng* 1985, 32, 230–236. [PubMed: 3997178]
27. Liu C; Zhao L; Tang H; Li Q; Wei S; Li J Life-threatening false alarm rejection in ICU: Using the rule-based and multi-channel information fusion method. *Physiol. Meas* 2016, 37, 1298–1312. [PubMed: 27454710]
28. Silva I; Moody B; Behar J; Johnson A; Oster J; Clifford GD; Moody GB Robust detection of heart beats in multimodal data. *Physiol. Meas* 2015, 36, 1629–1644. [PubMed: 26217894]
29. Behar J; Johnson A; Clifford GD; Oster J A comparison of single channel fetal ECG extraction methods. *Ann. Biomed. Eng* 2014, 42, 1340–1353. [PubMed: 24604619]
30. Gierałtowski J; Grzegorzczak I; Ciuchci ski K; Ko na K; Soli ski M; Podziemski P Algorithm for life-threatening arrhythmias detection with reduced false alarms. In Proceedings of the 2015 Computing in Cardiology Conference (CinC), Nice, France, 6–9 September 2015; pp. 1201–1204.
31. Krasteva V; Jekova I; Leber R; Schmid R; Abächerli R Superiority of classification tree versus cluster, fuzzy and discriminant models in a heartbeat classification system. *PLoS ONE* 2015, 10, e0140123. [PubMed: 26461492]
32. Rooijackers MJ; Rabotti C; Oei SG; Mischi M Low-complexity R-peak detection for ambulatory fetal monitoring. *Physiol. Meas* 2012, 33, 1135–1150. [PubMed: 22735075]
33. Gierałtowski J; Ciuchci ski K; Grzegorzczak I; Ko na K; Soli ski M; Podziemski P RS slope detection algorithm for extraction of heart rate from noisy, multimodal recordings. *Physiol. Meas* 2015, 36, 1743–1761. [PubMed: 26218763]
34. Sedghamiz H Matlab Implementation of Pan Tompkins ECG QRS Detector. Code Available at the File Exchange Site of MathWorks. 2014 Available online: <https://fr.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector> (accessed on 25 February 2019).
35. Antink CH; Leonhardt S; Walter M Reducing false alarms in the ICU by quantifying self-similarity of multimodal biosignals. *Physiol. Meas* 2016, 37, 1233–1252. [PubMed: 27454256]
36. Kalidas V; Tamil LS Cardiac arrhythmia classification using multi-modal signal analysis. *Physiol. Meas* 2016, 37, 1253. [PubMed: 27454417]
37. Sadr N; Huvanandana J; Nguyen DT; Kalra C; McEwan A; de Chazal P Reducing false arrhythmia alarms in the ICU using multimodal signals and robust QRS detection. *Physiol. Meas* 2016, 37, 1340. [PubMed: 27455121]
38. Plesinger F; Klimes P; Halamek J; Jurak P Taming of the monitors: Reducing false alarms in intensive care units. *Physiol. Meas* 2016, 37, 1313–1325. [PubMed: 27454821]
39. Khadra L; Al-Fahoum AS; Al-Nashash H Detection of life-threatening cardiac arrhythmias using the wavelet transformation. *Med. Biol. Eng. Comput* 1997, 35, 626–632. [PubMed: 9538538]

40. Christov II Real time electrocardiogram QRS detection using combined adaptive threshold. *Biomed. Eng. Online* 2004, 3, 28. [PubMed: 15333132]
41. Arzeno NM; Deng ZD; Poon C-S Analysis of First-Derivative Based QRS Detection Algorithms. *IEEE Trans. Biomed. Eng* 2008, 55, 478–484. [PubMed: 18269982]
42. Mjihad A; Rosado-Muñoz A; Bataller-Mompeán M; Francés-Villora JV; Guerrero-Martínez JF Ventricular Fibrillation and Tachycardia detection from surface ECG using time-frequency representation images as input dataset for machine learning. *Comput. Methods Programs Biomed* 2017, 141, 119–127. [PubMed: 28241963]
43. Prabhakararao E; Manikandan MS Efficient and robust ventricular tachycardia and fibrillation detection method for wearable cardiac health monitoring devices. *Healthc. Technol. Lett* 2016, 3, 239–246. [PubMed: 27733933]
44. Fallet S; Yazdani S; Vesin JM A multimodal approach to reduce false arrhythmia alarms in the intensive care unit. In *Proceedings of the 2015 Computing in Cardiology Conference (CinC)*, Nice, France, 6–9 September 2015; pp. 277–280.
45. Chen S; Thakor NV; Mower MM Ventricular fibrillation detection by a regression test on the autocorrelation function. *Med. Biol. Eng. Comput* 1987, 25, 241–249. [PubMed: 3329694]
46. Balasundaram K; Masse S; Nair K; Farid T; Nanthakumar K; Umopathy K Wavelet-based features for characterizing ventricular arrhythmias in optimizing treatment options. In *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, USA, 30 August–3 September 2011.
47. Li H; Han W; Hu C; Meng MQ-H Detecting ventricular fibrillation by fast algorithm of dynamic sample entropy. In *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, 19–23 December 2009.
48. Alonso-Atienza F; Morgado E; Fernandez-Martinez L; Garcia-Alberola A; Rojo-Alvarez JL Detection of Life-Threatening Arrhythmias Using Feature Selection and Support Vector Machines. *IEEE Trans. Biomed. Eng* 2014, 61, 832–840. [PubMed: 24239968]
49. Anas E; Lee SY; Hasan MK Sequential algorithm for life threatening cardiac pathologies detection based on mean signal strength and EMD functions. *Biomed. Eng. Online* 2010, 9, 43. [PubMed: 20815909]
50. Asadi F; Mollakazemi MJ; Ghiasi S; Sadati SH Enhancement of life-threatening arrhythmia discrimination in the intensive care unit with morphological features and interval feature extraction via random forest classifier. In *Proceedings of the 2016 Computing in Cardiology Conference (CinC)*, Vancouver, BC, Canada, 11–14 September 2016; pp. 57–60.
51. Eerikäinen LM; Vanschoren J; Rooijackers MJ; Vullings R; Aarts RM Reduction of false arrhythmia alarms using signal selection and machine learning. *Physiol. Meas* 2016, 37, 1204–1216. [PubMed: 27454128]
52. Gajowniczek K; Orłowski A; Zbkowski, T. Entropy Based Trees to Support Decision Making for Customer Churn Management. *Acta Phys. Pol. A* 2016, 129, 971–979.
53. Li HB; Wang W; Ding HW; Dong J Trees weighting random forest method for classifying high-dimensional noisy data. In *Proceedings of the 2010 IEEE 7th International Conference on E-Business Engineering*, Shanghai, China, 10–12 November 2010; pp. 160–163.
54. Gajowniczek K; Orłowski A; Zbkowski T Simulation Study on the Application of the Generalized Entropy Concept in Artificial Neural Networks. *Entropy* 2018, 20, 249.
55. Friedman J; Hastie T; Tibshirani R *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2001.
56. Mehrabi N; Morstatter F; Saxena N; Lerman K; Galstyan A A survey on bias and fairness in machine learning. *arXiv* 2019, arXiv:1908.09635
57. Hocking T *WeightedROC: Fast, Weighted ROC Curves*. R package Version 2018.10.1. 2018. Available online: <https://CRAN.R-project.org/package=WeightedROC> (accessed on 10 October 2019).
58. Roszkowska E Rank ordering criteria weighting methods—A comparative overview, *Optimum. Studia Ekon.* 2013, 5.

59. Stillwell WG; Seaver DA; Edwards W A comparison of weight approximation techniques in multiattribute utility decision making. *Organ. Behav. Hum. Perform* 1981, 28, 62–77.
60. The R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2014.
61. Wright MN; Ziegler A ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *J. Stat. Softw* 2017, 77.
62. Youden WJ An index for rating diagnostic tests. *Cancer* 1950, 3, 32–35. [PubMed: 15405679]
63. Kuhn M Building Predictive Models in R using the caret Package. *J. Stat. Softw* 2008, 28.
64. Gajowniczek K; Karpio K; Łukasiewicz P; Orłowski A; Z bkowski T Q-Entropy Approach to Selecting High Income Households. *Acta Phys. Pol. A* 2015, 127, A-38–A-44.
65. Breiman L; Friedman JH; Olshen RA; Stone CJ *Classification and Regression Trees*; Wadsworth Statistics/Probability Series; CRC Press: Boca Raton, FL, USA, 1984.
66. Gajowniczek K; Z bkowski T; Sodenkamp M Revealing Household Characteristics from Electricity Meter Data with Grade Analysis and Machine Learning Algorithms. *Appl. Sci* 2018, 8, 1654.
67. Pohlert T The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR). R Package. Available online: <https://cran.r-project.org/web/packages/PMCMR/vignettes/PMCMR.pdf> (accessed on 29 December 2019).



**Figure 1.** Weights distribution in terms of number of trees and value of the parameter  $p$ .



**Figure 2.**  
Nemenyi diagram for benchmarking methods for Score measure.

**Table 1.**

Weighting methods for ensemble classifiers in the literature.

Work	Method Applied	Conclusion
[11]	Tree-level weights in Random Forest.	Method does not dramatically improve predictive ability in high-dimensional genetic data, but it may improve performance in other domains.
[14]	Tunable weighted bagged ensemble using CART, Naïve Bayes, KNN, SVM, ANN and Logistic Regression.	Approach can usually outperform pure bagging, however, there are some cons in terms of time considerations in effectively choosing tunable parameters aside from a grid search.
[15]	Variable importance-weighted Random Forest.	Better prediction power in comparison to existing random forests granting the same weight to all tree models.
[16]	Refined weighted Random Forest (assigning different weights to different decision trees).	Better prediction power in comparison to standard random forests due to the following: (1) all training data including in-bag data and Out-of-Bag data is used and (2) the margin between probability of predicting true class and false class label applied.
[20]	Optimality conditions for four combination methods: majority vote (MV), weighted majority vote (WMV), the recall combiner (REC) and Naive Bayes (NB).	Experiments revealed that there is no dominant combiner. NB was the most successful but the differences with MV and WMV were not found to be statistically significant.
[22]	Weighting each tree by replacing the regular average with a Cesaro average (CRF—Cesaro Random Forest).	Although the Cesaro random forest appears to be competitive to the classical RF, it has limitations i.e., the way to determine the sequencing of trees (what impacts the results) and the probability estimates of class membership are not available.
[23]	Variable performance-weighted and Recency-weighted random forests.	The results show that recency-weighted ensembles of random forests produce superior results in terms of both profitability and prediction accuracy compared with other ensemble techniques.
[24]	Weighted random survival forest by assigning weights to survival decision trees or to their subsets.	Numerical examples with real data illustrate the outperformance of the proposed model in comparison with the original random survival forest.

**Table 2.**

Methods of detecting arrhythmias.

Arrhythmia/ Complex	Method	Work
QRS Detection	Pan-Tompkins (filtering techniques); Threshold-based detection; Multimodal data methods; Gradient calculations; Based on Peak energy; Markov-model; RS Slope detection; Low-complexity R-peak detector.	[26–35]
Asystole	Short term autocorrelation analysis; Flat line artefacts definition; Frequency domain analysis; Signal quality based rules.	[35–37]
Bradycardia and Tachycardia	Threshold +Support vector machine; Beat-to-beat Correlogram 2D.	[35,36]
Ventricular Tachycardia	Time-frequency representation images; Spectral characteristics of ECG; Spectra purity index; Autocorrelation function.	[31,36,38–44]
Ventricular Flutter or Fibrillation	Autocorrelation analysis; Wavelet transformations; Sample entropy; Machine learning methods with features derived from signal morphology and analysis of power spectrum; Time-frequency representation images; Empirical mode decomposition; The zero crossing rate combined with base noise suppression with discrete cosine transform and beat-to-beat intervals.	[39,42,43,45–49]
All types	Rule based methods; Regular-activity test; Single- and multichannel fusion rules; Machine learning algorithms; SVM—Support Vector Machines; LDA—Linear discriminant analysis; Random Forest classifiers.	[27,35,38,50,51]

**Table 3.**

An example of weights deriving.

Tree No.	AUC <sub>INB</sub>	AUC <sub>OOB</sub>	Equation (4) ( $\alpha=0.5$ )	Ranking	Nominator ( $p = 2$ )	Final We ights
1	0.70	0.70	0.350	3	4	0.133
2	0.65	0.55	0.325	4	1	0.034
3	0.90	0.80	0.450	1	16	0.533
4	0.85	0.80	0.425	2	9	0.300

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript



**Table 4.**

Improvement of Area Under the Curve (AUC) and Score for Asystole on validation sample in terms of the  $\alpha$ -parameter and  $p$ -parameter.

	Base	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
AUC = 0.93	0	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050	0.050
	0.1	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.2	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.3	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.4	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.5	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.6	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.050
	0.7	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055
	0.8	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055
	0.9	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.060	0.060
1	0.000	0.055	0.055	0.055	0.055	0.055	0.055	0.060	0.060	0.060	0.060	
SCORE = 61.75	0	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.1	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.2	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.3	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.4	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.5	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.6	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.7	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.8	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
	0.9	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00
1	0.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	19.00	

**Table 5.**

Improvement of AUC and Score for Bradycardia on validation sample in terms of the  $\alpha$ -parameter and  $\rho$ -parameter.

	Base	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
AUC = 0.95	<b>0</b>	0.000	0.000	-0.005	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011
	<b>0.1</b>	0.000	0.000	-0.005	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011
	<b>0.2</b>	0.000	0.000	-0.005	-0.005	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011
	<b>0.3</b>	0.000	0.001	-0.005	-0.005	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011	-0.011
	<b>0.4</b>	0.000	0.001	-0.005	-0.005	-0.005	-0.011	-0.011	-0.011	-0.005	-0.005	-0.005
	<b>0.5</b>	0.000	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005
	<b>0.6</b>	0.000	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005
	<b>0.7</b>	0.000	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005
	<b>0.8</b>	0.000	0.001	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005
	<b>0.9</b>	0.000	0.001	0.001	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005
<b>1</b>	0.000	0.001	0.001	0.001	0.001	-0.005	-0.005	-0.005	-0.005	-0.005	-0.005	
SCORE = 77.73	<b>0</b>	0.00	-0.02	-0.02	-0.02	-0.02	-1.27	-1.27	-1.27	-1.27	-1.27	-1.27
	<b>0.1</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-1.27	-1.27	-1.27	-1.27	-1.27
	<b>0.2</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-1.27	-1.27	-1.27	-1.27	-1.27
	<b>0.3</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-1.27	-1.27	-1.27	-1.27
	<b>0.4</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-1.27	-1.27	-1.27
	<b>0.5</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-1.27
	<b>0.6</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
	<b>0.7</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
	<b>0.8</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
	<b>0.9</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
<b>1</b>	0.00	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 6.**

Improvement of AUC and Score for Tachycardia on validation sample in terms of the  $\alpha$ -parameter and  $p$ -parameter.

	Base	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
AUC = 0.99	<b>0</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	<b>0.1</b>	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008	-0.008	-0.008
	<b>0.2</b>	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008	-0.008	-0.008
	<b>0.3</b>	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008	-0.008
	<b>0.4</b>	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008	-0.008
	<b>0.5</b>	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.001	-0.001
	<b>0.6</b>	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.001	-0.001
	<b>0.7</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.001
	<b>0.8</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008
	<b>0.9</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	-0.008
<b>1</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.008	-0.008	
SCORE = 81.08	<b>0</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.1</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.2</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.3</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.4</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.5</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.6</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-9.25
	<b>0.7</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.8</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	<b>0.9</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<b>1</b>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

**Table 7.**

Improvement of AUC and Score for Ventricular Fibrillation or Flutter on validation sample in terms of the  $\alpha$ -parameter and  $p$ -parameter.

	Base	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
AUC = 0.97	<b>0</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.1</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.2</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.3</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.4</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.5</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.6</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.7</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.8</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
	<b>0.9</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030
<b>1</b>	0.000	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.009
SCORE = 30.56	<b>0</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	55.55	55.55	55.55
	<b>0.1</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.2</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.3</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.4</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.5</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.6</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.7</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.8</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
	<b>0.9</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51
<b>1</b>	0.00	55.55	55.55	55.55	55.55	55.55	55.55	55.55	54.51	54.51	54.51	

**Table 8.**

Improvement of AUC and Score for Ventricular Tachycardia on validation sample in terms of the  $\alpha$ -parameter and  $p$ -parameter.

	Base	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
AUC = 0.87	<b>0</b>	0.000	-0.001	0.000	0.003	0.001	0.002	0.001	0.001	0.001	0.001	0.003
	<b>0.1</b>	0.000	-0.001	0.000	0.002	0.002	0.002	0.002	0.002	0.001	0.000	0.002
	<b>0.2</b>	0.000	-0.002	0.000	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.002
	<b>0.3</b>	0.000	-0.002	0.001	0.001	0.002	0.002	0.001	0.001	0.001	0.001	0.003
	<b>0.4</b>	0.000	-0.002	0.001	0.000	0.002	0.002	0.001	0.001	0.000	0.001	0.003
	<b>0.5</b>	0.000	-0.002	0.001	0.000	0.001	0.002	0.002	0.002	0.001	0.001	0.002
	<b>0.6</b>	0.000	-0.002	0.001	0.000	0.001	0.002	0.002	0.001	0.002	0.002	0.003
	<b>0.7</b>	0.000	-0.002	0.000	0.000	0.001	0.002	0.002	0.002	0.001	0.001	0.003
	<b>0.8</b>	0.000	-0.002	0.001	0.000	0.001	0.001	0.002	0.002	0.001	0.003	0.002
	<b>0.9</b>	0.000	-0.002	0.000	0.001	0.001	0.001	0.002	0.001	0.000	0.002	0.002
<b>1</b>	0.000	-0.002	-0.001	0.000	0.001	0.001	0.002	0.001	0.000	0.001	0.001	
SCORE = 31.54	<b>0</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.31	0.31	0.21	0.21	0.21
	<b>0.1</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.31	0.21	0.21	0.21	0.21
	<b>0.2</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.42	0.31	0.21	0.21
	<b>0.3</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.42	0.31	0.31	0.31
	<b>0.4</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.31	0.31	0.31	0.31
	<b>0.5</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.42	0.31	0.31	0.31
	<b>0.6</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.31	0.31	0.31	0.31
	<b>0.7</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	0.31	1.01	1.01	0.33
	<b>0.8</b>	0.00	-0.26	1.11	0.42	0.42	0.42	0.42	1.01	1.01	0.33	0.33
	<b>0.9</b>	0.00	-0.26	0.44	0.42	0.42	0.42	1.11	1.01	0.33	0.33	0.33
<b>1</b>	0.00	-0.26	0.44	0.42	0.42	1.11	1.11	0.33	0.33	0.23	0.23	

**Table 9.**

Detailed classification results for benchmarking methods.

Arrhythmia Type	Method	AUC	Score
Asystole	Weighted RF ( $\alpha = 0.9, p = 4.5$ )	(98.5 $\pm$ 3.1)	(80.7 $\pm$ 8.7)
	Standard RF	(92.5 $\pm$ 3.5)	(61.7 $\pm$ 9.2)
	CART ( $cp = 0.065$ )	(86.0 $\pm$ 4.2)	(52.1 $\pm$ 10.9)
	C-SVM (polynomial = 2, $\epsilon = 0.3, C = 0.4$ )	(92.0 $\pm$ 3.7)	(61.7 $\pm$ 9.2)
	AdaBoost	(91.9 $\pm$ 3.9)	(61.7 $\pm$ 9.2)
Extreme Bradycardia	Weighted RF ( $\alpha = 0.5, p = 0.5$ )	(95.6 $\pm$ 4.4)	(77.7 $\pm$ 9.7)
	Standard RF	(95.0 $\pm$ 4.5)	(77.7 $\pm$ 9.7)
	CART ( $cp = 0.083$ )	(87.5 $\pm$ 4.4)	(63.1 $\pm$ 10.5)
	C-SVM (polynomial = 2, $\epsilon = 0.3, C = 0.4$ )	(95.2 $\pm$ 4.5)	(77.7 $\pm$ 9.7)
	AdaBoost	(95.1 $\pm$ 4.6)	(77.7 $\pm$ 9.7)
Ventricular Tachycardia	Weighted RF ( $\alpha = 0.7, p = 5.0$ )	(87.5 $\pm$ 3.5)	(31.9 $\pm$ 2.7)
	Standard RF	(87.3 $\pm$ 3.5)	(31.5 $\pm$ 2.7)
	CART ( $cp = 0.011$ )	(72.6 $\pm$ 4.2)	(27.7 $\pm$ 4.2)
	C-SVM (radial = 0.1, $\epsilon = 0.1, C = 0.8$ )	(86.1 $\pm$ 3.7)	(30.5 $\pm$ 3.7)
	AdaBoost	(83.1 $\pm$ 3.9)	(29.9 $\pm$ 3.7)
Ventricular Fibrillation or Flutter	Weighted RF ( $\alpha = 0.5, p = 1.0$ )	(99.9 $\pm$ 0.1)	(86.1 $\pm$ 7.7)
	Standard RF	(97.0 $\pm$ 2.1)	(30.6 $\pm$ 13.9)
	CART ( $cp = 0.017$ )	(89.9 $\pm$ 8.7)	(29.4 $\pm$ 16.6)
	C-SVM (radial = 0.5, $\epsilon = 0.1, C = 0.8$ )	(97.5 $\pm$ 5.3)	(50.1 $\pm$ 12.8)
	AdaBoost	(97.5 $\pm$ 5.3)	(50.1 $\pm$ 12.8)
Extreme Tachycardia	Weighted RF ( $\alpha = 0.1, p = 1.0$ )	(99.2 $\pm$ 0.1)	(81.1 $\pm$ 7.7)
	Standard RF	(99.2 $\pm$ 0.1)	(81.1 $\pm$ 7.7)
	CART ( $cp = 0.090$ )	(64.2 $\pm$ 8.6)	(53.6 $\pm$ 9.9)
	C-SVM (polynomial = 3, $\epsilon = 0.5, C = 0.6$ )	(99.2 $\pm$ 0.1)	(81.1 $\pm$ 7.7)
	AdaBoost	(99.2 $\pm$ 0.1)	(81.1 $\pm$ 7.7)