Blind Source Separation for the Aggregation of Machine Learning Algorithms: An Arrhythmia Classification Case

Krzysztof Gajowniczek 1,*, Iga Grzegorczyk 2, Michał Gostkowski 3 and Tomasz Ząbkowski 1

1 Department of Artificial Intelligence, Institute of Information Technology, Warsaw University of Life Sciences SGGW, 02-776 Warsaw, Poland; tomasz_zabkowski@sggw.pl
2 Department of Physics of Complex Systems, Faculty of Physics, Warsaw University of Technology, 00-662 Warsaw, Poland; igagrz@gmail.com
3 Department of Econometrics and Statistics, Institute of Economics and Finance, Warsaw University of Life Sciences SGGW, 02-776 Warsaw, Poland; michal_gostkowski@sggw.pl

* Correspondence: krzysztof_gajowniczek@sggw.pl

Received: 3 February 2020; Accepted: 29 February 2020; Published: 3 March 2020

Abstract: In this work, we present an application of the blind source separation (BSS) algorithm to reduce false arrhythmia alarms and to improve the classification accuracy of artificial neural networks (ANNs). The research was focused on a new approach for model aggregation to deal with arrhythmia types that are difficult to predict. The data for analysis consisted of five-minute-long physiological signals (ECG, BP, and PLETH) registered for patients with cardiac arrhythmias. For each patient, the arrhythmia alarm occurred at the end of the signal. The data present a classification problem of whether the alarm is a true one—requiring attention or is false—should not have been generated. It was confirmed that BSS ANNs are able to detect four arrhythmias—asystole, ventricular tachycardia, ventricular fibrillation, and tachycardia—with higher classification accuracy than the benchmarking models, including the ANN, random forest, and recursive partitioning and regression trees. The overall challenge scores were between 63.2 and 90.7.

Keywords: arrhythmia; false alarm; blind source separation; machine learning

1. Introduction

Cardiac arrhythmia refers to a condition in which the heart muscle does not contract in a regular way. In this context, it cannot efficiently pump blood to the brain, lungs, and other organs, affecting their functioning or even damaging them. Within the scope of this article are the five most common types of cardiac arrhythmias. Three of them are diagnosed on the basis of physiologically incorrect lengths of intervals between consecutive heart muscle contractions. One is asystole, where a patient’s heart stops contracting and is inactive for at least 4 s. Another is bradycardia, where the heart rate is unnaturally slow—even lower than 40 beats per minute (bpm). Another is tachycardia, characterized by an unphysiologically fast pace of heart contractions while at rest. The shape of QRS complexes (the combination of three of the graphical deflections seen on a typical electrocardiogram), which represent the depolarization of ventricles, remains mostly unchanged, but the pace of heart contractions in this arrhythmia might exceed 140 bpm. Another type of tachycardia (ventricular tachycardia) occurs when additional electrical impulses are created in ventricles, which causes the heart rate to accelerate. Moreover, in an electrocardiogram (ECG), there is no longer a visible P-wave responsible for atrial contractions, and the signal consists only of unnaturally widened QRS complexes, which may be missed by regular QRS detection algorithms. Ventricular tachycardia can evolve into ventricular flutter/fibrillation (VF), the most dangerous arrhythmia for cardiovascular system functioning. In this
state, heart muscle contractions occur in a chaotic, irregular way. From a diagnostic point of view, in this case it is impossible to distinguish singular QRS complexes, and ECG signals take the form of oscillatory waveforms. This has to last for at least 4 s to be diagnosed as VF.

To identify and analyze asystole, bradycardia, and tachycardia, the key is to properly determine the locations of QRS complexes (or the lack of them, in the case of asystole). The ECG signal is the key source of information about a patient’s cardiovascular condition, but as the measurements are gathered from a human patient’s artefacts related to subject/electrode movement, sweating or muscle contractions might occur in the signal. That is why special attention needs to be paid to data quality assessment. In bedside monitors apart from ECG, there are recorded pulsatile waveforms, including blood pressure (BP) or photoplethysmogram (PLETH) waveforms. These recordings may deliver additional information about the heart rate when ECG signal is of poor quality. In Table 1, we provide commonly used methods for QRS complex detection from ECG signals only, as well as works describing algorithms enhancing these standard methods with information obtained from pulsatile waveforms. There is also a segment dedicated to each arrhythmia type and corresponding methods of diagnosing those. The last row of the table includes general methods applicable to classification of different arrhythmia types.

### Table 1. Methods of detecting arrhythmias.

| Method | Work |
|--------|------|
| QRS Detection | • Based on peak energy  
• Multimodal data methods  
• Markov model  
• Gradient calculations  
• RS slope detection  
• Low-complexity R-peak detector  
• Threshold-based detection  
• Gradient calculations  
• Pan-Tompkins (filtering techniques) |
| Arrhythmia type: | |
| Bradycardia | • Beat-to-beat correlogram 2D  
• Threshold + support vector machine (SVM) |
| Tachycardia | • Beat-to-beat correlogram 2D  
• Threshold + SVM |
| Asystole | • Frequency domain analysis  
• Flat line artefacts definition  
• Signal-quality-based rules  
• Short-term autocorrelation analysis |
| Ventricular tachycardia | • Spectra purity index  
• Spectral characteristics of ECG  
• Autocorrelation function  
• Time-frequency representation images |
With the onset of the artificial intelligence (AI) era, blind source separation (BSS) methods are gaining significant interest nowadays. BSS has proven a great potential in multiple practical areas, e.g., wireless communication systems, text analysis, seismic monitoring, signal separation, stock prediction, and image and biomedical signal processing [27,28].

A number of novel algorithms related to BSS have been developed recently, and these have played an important role in many disciplines [28]. These algorithms can be systemized in several ways taking into account the source separation condition or the restrictions to the source features, including independent component analysis (ICA) [29], sparse component analysis (SCA) [30], nonnegative matrix factorization (NMF) [31], and bounded component analysis (BCA) [32].

Aggregation of models based on machine learning algorithms is usually performed by supervised learning of ensemble models [33]. The ensemble methods aim at combining forecasts of several basic models in order to improve accuracy with respect to one basic model [34]. The most common approaches to combining forecasts from basic models are as follows:

1) Averaging is designed to build several models (usually of the same type) from different samples of the training dataset. The main idea is to create multiple models independently (e.g., bagging methods [35] or random forests [36]) and then combining their forecasts by averaging.

2) Boosting is aimed at building multiple models (also typically of the same type), where each new model is focused on fixing the prediction errors of a prior model in the sequence of models (e.g., AdaBoost [37] or gradient tree boosting [38]). Base estimators are built sequentially, and one tries to reduce the bias of the combined estimator from previous steps.

3) Voting is aimed at building multiple models (typically of differing types) based on the same training dataset, and then simple statistics, like the mean, are used to combine predictions [35].

Artificial neural networks (ANNs) are powerful and extensive learning models applicable to various domains. They have been successfully applied in areas such as system control, pattern recognition, and signal processing to solve numerous industrial problems [39–43]. ANNs are very accurate in certain practical applications because of several features. First, they possess an ability to generalize with a high tolerance for incomplete or noisy data. Second, neural networks do not require any a priori assumptions about the distribution of the data. Third, they are universal for modeling nonlinear behavior.

| Method | Work |
|--------|------|
| Ventricular flutter or fibrillation | Machine learning methods with features derived from signal morphology and analysis of power spectrum |
| | Wavelet transformations |
| | Empirical mode decomposition |
| | Sample entropy |
| | Time-frequency representation images |
| | Autocorrelation analysis |
| | The zero crossing rate combined with base noise suppression with discrete cosine transform and beat-to-beat intervals |
| All types | Single- and multichannel fusion rules |
| | Regular activity test |
| | Rule-based methods |
| | Machine learning algorithms: [2,10,13,25,26] |
| | Linear discriminant analysis (LDA) |
| | SVMs |
| | Random forest classifiers |
In this paper, using the PhysioNet Challenge 2015 dataset gathered for subjects with five different arrhythmia types, we aimed to improve the classification of arrhythmias using BSS methods applied to neural networks (BSS ANNs). We assume that every classification result includes latent components with both constructive and destructive characteristics. The elimination of the destructive components should improve the accuracy of the models. The remaining constructive components are remixed in a nonlinear adaptive system represented by a multilayer perceptron (MLP) neural network.

To prove the effectiveness of the approach we compared the proposed BSS approach with other benchmarking methods, including artificial neural networks (ANN), random forests (RF), and recursive partitioning and regression trees (RPART).

The remainder of this paper is structured as follows: Section 2 delivers an overview of the BSS approach applied to neural networks. In Section 3, the design of the experiment was outlined, including the details of numerical implementation, description of the feature (variable) vector, model performance measuring, and benchmarking methods. Section 4 outlines the experiments and discusses the results. The concluding remarks are provided in Section 5.

2. Neural Blind Source Separation Aggregation

2.1. Standard Blind Source Separation

The blind separation of signals assumes that there is a set of signals generated by specific sources, and it is then mixed in a certain system. It is assumed that both source signals and the mixing system itself are unknown. Source signals may include both significant and undesired components (e.g., noise and interference). By blind separation of signals, we mean the reproduction of source signals based only on the observed signals. The simplest approach assumes a linear model (system) of mixing signals, defined by the formula:

\[ x(t) = As(t), \quad (1) \]

where \( t \) is an observation number or time index, \( s(t) = [s_1(t), \ldots, s_n(t)]^T \) is a vector of source (unknown) signals, \( x(t) = [x_1(t), \ldots, x_m(t)]^T \) is a vector of the observed signals, and \( A \in \mathbb{R}^{m \times n} \) is an unknown nonsingular matrix representing the mixing system. In order to solve the above formula, some assumptions are usually made, such as the characteristics of the source signals, that the columns of the \( A \) matrix are linearly independent, or that the number of source signals is equal to the number of the observed signals, \( m = n \).

Due to the aforementioned assumptions and ambiguities, it is not possible to obtain an ideal solution without a priori knowledge about \( A \) and \( s(t) \). Therefore, the purpose of the blind signal separation is to find (estimate and reconstruct) a separation matrix \( W \) such that the estimation of source signals can be described by the formula:

\[ y(t) = WX(t) = PDs(t), \quad (2) \]

where \( P \) is a permutation matrix defining the order of estimated signals, \( D \) is a diagonal scaling matrix, and \( y(t) \) is a vector of estimated signals. Therefore, the correct solution in the blind separation of signals is the reproduction of original signals, which are scaled and occur in a different order than the source signals.

In practice, the way in which BSS is achieved depends on the real and underlying characteristics of the source signals, such as the smoothness, decorrelation, statistical independence, non-negativity, sparsity, and non-stationarity. Many analytical methods explore different properties of the data, and the choice of a particular method depends on the nature of the specific problem and data characteristics.

The independent component analysis (ICA) for BSS assumes that the source vector \( s(t) \) in a model (Equation (1)) has mutually independent components. In consequence, the mixing matrix \( A \) in (Equation (1)) is not well-defined, so some extra assumptions have to be made [44]; i.e., the source
components are mutually independent, \( E(s(t)) = 0 \), and \( \mathbb{E}(s(t)^T s(t)) = I \), at most one of the components is Gaussian, and each source component is independent and identically distributed.

One of the most classical ICA algorithms is the joint approximate diagonalization of eigenmatrices (JADE). It utilizes joint diagonalization when estimating the matrix \( W \) and fourth-order moments [44]. Gaussian distributions have zero excess kurtosis, but the canonical assumption of ICA is non-Gaussianity. To estimate source vectors JADE seeks for an orthogonal rotation of the observed mixed vectors, as these possess high values of excess kurtosis.

2.2. Blind Source Separation Aggregation

In our approach, we assume that results generated by any classification model usually consists of hidden components of both constructive and destructive types. For a few models, some of the components can be common to all of them. With this assumption our goal is to find the common components and distinguish those with a constructive influence on the classification accuracy from the destructive components. Therefore, the starting point of BSS aggregation is an assumption that there are a set of prediction results for any class of models \( x_i(t), \ i = 1, \ldots, m \). The results are stacked into multidimensional variable \( x(t) = [x_1(t), \ldots, x_m(t)]^T \) to determine whether the results are good for further analysis. The forecasted values, to some extent, have to correspond to the observed values, and they also have to differ to some extent. Therefore, it can be stated that a given prediction result of the model is a combination of constructive components \( \tilde{s}_j(t), \ j = 1, \ldots, p \) related to similarity and destructive components \( s'_{l}(t), \ l = 1, \ldots, q \) related to differences in predicted and observed values. All these components can be treated as hidden base components contained in a multidimensional \( s \)-variable. Hence, it can be stated that the vector \( s \) with source signals contains:

\[
\begin{bmatrix}
\tilde{s}_1(t), \ldots, \tilde{s}_p(t), \ s'_{p+1}(t), \ldots, s'_{p+q}(t)
\end{bmatrix}^T.
\]

Once the separation of latent (hidden) components completes, the destructive components can be removed and replaced with zeros \( (s_l = 0) \) to obtain an improved version \( \hat{x}(t) \) of the real (observed) signals \( x(t) \):

\[
\hat{x}(t) = A\begin{bmatrix}
\tilde{s}_1(t), \ldots, \tilde{s}_p(t), 0_{p+1}(t), \ldots, 0_{p+q}(t)
\end{bmatrix}^T.
\]

The removal of the destructive signal in (Equation (4)) is equivalent to providing zero in the respective column of \( A \). Therefore, if the mixing matrix is expressed as \( A = [a_1, \ldots, a_n] \), then the improved results can be formulated as follows (the marker denoted as ~ is further inherited from constructive components):

\[
\hat{x}(t) = A\tilde{s}(t) = \tilde{A}s(t),
\]

where \( \tilde{A} = [a_1, \ldots, a_p, 0_{p+1}, \ldots, 0_n] \). The improved prediction results, due to the noise elimination can be rewritten as a linear combination of the base results:

\[
\hat{x}(t) = \tilde{A}Wx(t).
\]

Thus, noise filtration with BSS approach can be assumed as a form of aggregation.

2.3. Neural Mixing System

The components can be neither purely constructive nor purely destructive and, therefore, their impact might have a weight other than 0. Moreover, a respective component might have a constructive effect on one model and a destructive effect on another. There might also be components that are destructive when they are single but constructive when considered in a pair or in a group [45]. This means that it might exists a better mixing system than the one described by \( \tilde{A} \), that uses simple
linear relationship. The wide range of nonlinear mixing systems can be modeled similarly to neural networks systems [45]; therefore, a brief introduction is given here.

ANNs are computational systems inspired by biological neurons connected in networks that are processing the signals (e.g., animals’ brains). Based on examples the ANN systems learn tasks, usually without task-specific programming [41]. Most implementations of an ANN assume the synapse signal is a real number, and the output of each processing neuron is derived from a non-linear activation function, being the sum of its input signals. The connections between the neurons and synapses have the weights that adapt as the learning progresses. The weights are to increase or decrease the signal power of the synapses [41].

Usually neurons are organized into the layers with different types of input transformations [41]. Signals move from the first (input) layer to the output, possibly going through multiple layers. Figure 1 presents an example of a feedforward ANN with three input neurons (representing three features), three neurons in hidden layer and the output neuron that represents the target variable. Assuming an ANN with one hidden layer and $J$ hidden neurons in that layer, and $m$ input neurons in that structure then the following function needs to be calculated (please compare the formula with Figure 1) [41]:

$$\text{ANN} = f \left( w_0 + \sum_{j=1}^{J} w_j \left( w_{0j} + \sum_{i=1}^{m} w_{ij} z_i \right) \right), \quad (7)$$

where $w_0$ is the synaptic weight of the intersection point of the output neuron, $w_{0j}$ is the synaptic weight of the intersection point of the $j$-th hidden neuron, $z_i$ is the $i$-th feature, $w_{ij}$ is the weight corresponding to the synapse beginning with the $j$-th hidden neuron and leading to the output neuron. Finally, $w_{ij}$ is the weight of the $i$-th neuron from the input layer to the $j$-th hidden layer [41]. In addition, $f$ is an activation function that is typically limited to a non-linear, non-decreasing, and differentiated function, such as a sigmoid function.

**Figure 1.** Example of an artificial neural network (ANN) with three input variables, one hidden layer with three neurons inside and one output neuron (representing target variable).

Weights (parameters) of an ANN are adjusted by learning algorithms during the iterative training process, which stops if a predefined criterion is met (such as the number of iterations/epochs).
A common learning algorithm is a backpropagation algorithm modifying neural network weights to find a local (ideally, a global) minimum of a particular error function $E$, such as the sum of squares [41]. That is why the gradient of the error function is computed in line with the following equation (all the weights are kept in one vector with $(m + 1) \ast J + (J + 1)$ weights):

$$w_k^{(i+1)} = w_k^{(i)} - \eta_k \text{sign} \left( \frac{\partial E^{(i)}}{\partial w_k^{(i)}} \right)$$

where $k$ is an index of a particular weights, $i$ denotes the iteration step, and $\eta_k$ is the learning rate adjusted in the following way: to be increased if the corresponding partial derivative retains its sign, or to be decreased if the partial derivative of the error function changes its sign. When the sign changes that means that the minimum of the error function is missed due to a learning rate that is too high [41].

The new mixing system can be formulated as the neural system:

$$x(t) = f \left( w_0 + \sum_{j=1}^{J} w_j \left( \frac{\sum_{i=1}^{p+q} w_{ij} s(t_i) \right) \right)$$

According to Equation (9), it can be stated that the first weight layer will produce outputs associated with Equation (5) if we take \sum_{i=1}^{p+q} w_{ij} = 1 (please see Figure 2). Additionally, the neural mixing system uses some nonlinearities (e.g., sigmoid activation functions) and the second layer [45], which is why the mixing system gains some tolerance and flexibility in comparison to the linear form (Equation (5)). If the entire neural mixing structure starting from the system described by $\tilde{A}$ is taken in the first iteration of learning, it is expected that the final results will be improved (see Figure 3).

![Figure 2. Neural mixing system.](image-url)
As mentioned before, to identify asystole, bradycardia, and tachycardia, consecutive heartbeats needed to be properly located. Hence, the first step in creating the variables was QRS complex detection in the ECG signal, what was accomplished by a low-complexity R-peak detector [26]. Simultaneously, an open source \textit{wabp} algorithm [5] was used to localize the beats in the ABP and PLETH waveforms provided. Quality detection of the beats was assessed through the comparison of the QRS locations between the signals. When the beat was found in both signals then it was marked as a true positive (TP), otherwise it was treated either a false positive (FP) or false negative (FN) depending on the order in which the signals were compared [26,46]. Further, an F1-score was calculated as $F1 = \frac{2 TP}{2 TP + FP + FN}$. The closer the beat locations in the signals were, the higher F1-score was and closer to 1. In case of no matches of beat locations were observed the F1-score was assigned 0. Each time, two signals with the highest F1-score were considered for the analysis [26,46]. Hence, waveform signals were critical for assessing ECG signal quality. They are different physiological measures and hence are prone to different disturbances. If one of them shows symptoms of arrhythmia and the other one is normal, it is possible that the first one was of poor quality. Choosing high-quality signals at this step determines better variables for later predictions.

When diagnosing ventricular tachycardia and ventricular flutter or fibrillation, all the variables were prepared with a spectral purity approach (SPI) [19,26]. These arrhythmias require a different approach because, as mentioned before (see Section 1), the identification of physiological QRS complexes...
is unavailable due to the nature of the originated ventricular arrhythmias. Therefore, the minima and the maxima of the SPI were assumed as the variables to determine whether the alarm should have been generated—i.e., a max and min SPI for ventricular tachycardia and a max SPI for ventricular fibrillation or flutter.

3.2. Numerical Implementation

A numerical experiment was prepared with the use of R programming software [47] installed on a personal computer (Intel Core i7-9750H 2.6 GHz processor with 12 threads and 32 GB of RAM) with an Ubuntu 18.04 operating system. The entire neural BSS aggregation system was built based on author’s modification of the following R libraries: neuralnet—neural networks using backpropagation were applied, resilient backpropagation with or without weight backtracking, or a modified globally convergent version [48]; JADE—implementing Cardoso’s JADE algorithm as well as his functions for joint diagonalization and several other BSS methods, such as AMUSE and SOBI [44]. The optimal threshold to determine the class output was calculated based on the Youden index and using the pROC library [49].

The performance measures for the training and validation datasets were calculated using a $k$-fold cross-validation [46]. In particular, $k$ was set to 10 when there were more than 10 observations in a minority class. Otherwise, the number of $k$ was set to the quantity of a minority class to make sure that there was at least one observation per class (e.g., six folds for ventricular fibrillation or flutter) [46]. Each of the $k$ sets was created based on the stratified sampling using the createFolds function to make sure that the class distribution in each set represents the class distribution of the entire data set [50]. The summary of the results is presented as an average performance over $k$-fold (together with the standard errors of the estimates).

3.3. Performance Measures

For the models built with any statistical learning algorithm, a proper evaluation is crucial. Hence, various evaluation metrics were used in this research. The primary measure to address the Challenge Score system is the Score measure of the following form:

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

where in the case of a binary classification the following abbreviations are used [39]: $TN$ and $TP$ denote the number of appropriately indicated negative or positive observations, $FP$ denotes the observations predicted as Yes while the true target is No, and at the end $FN$ stands for the number of observations predicted as No while the true target is Yes. On the basis of the above formula, it can be concluded that the measure was designed in such a way that takes $FN$ (i.e., life-threatening events considered by the model irrelevant) especially seriously [5,46].

Another measure used in this experiment was the area under the ROC curve (AUC), which is particularly important since it was used to tune the parameters of each model [46]. For a binary classifier the AUC equals to the probability that the model places the randomly selected positive observation higher the randomly selected negative observation [39]. The AUC ranges from 0 to 1, where the higher the value, the better the accuracy of the model. Estimating AUC incorporates two indices (extension of these used in Equation (10)), i.e., the sensitivity, defined as $TP / (FN + TP)$, and the specificity, defined as $FP / (FP + TN)$. Since a binary classifier returns a class probability, the outcome has to be transformed into two possible outcomes (Yes or No) based on the threshold value (the default value is usually 0.5). Specificity and sensitivity are computed at various threshold values, such as $(0.00, 0.02, 0.04, \ldots, 1.00)$ [41].

Eventually, in order to determine the optimal threshold for each class, Youden’s $J$ statistic [51] was incorporated (as it is important for the Score measure). In short, the Youden index is the coordinate
on the ROC that is farthest from the diagonal line (random model); i.e., it maximizes the difference between sensitivity and specificity.

3.4. Benchmarking Methods

The following benchmarking algorithms have been used in the experiment. The first one is the CART algorithm which is an implementation of classification and regression trees [49]. It applies pruning during the growth stage what prevents new splits from being created when the previous splits deliver only small improvement of the accuracy. The complexity parameter \( cp \) varies in the analysis from 0 to 0.1 in increments of 0.01 [42].

The second one is the model that was built based on the framework described in [46]. Eventually, final results were derived based on the averaging over all raw models and were obtained in the second stage of the proposed approach (refer to Figure 3).

Benchmarking methods were implemented using the \texttt{rpart} library, incorporating recursive partitioning for classification, regression, and survival trees following the functionality presented in [52], and the \texttt{ranger} library [53], incorporating Breiman’s state-of-the-art RF algorithm.

4. Empirical Results

The classification accuracy for each method was assessed with the AUC and a challenge score (Equation (10)) within the training and the validation dataset. The results are presented in Figure 4. Each modeling method is shown in a different color, and the whiskers represent the standard error of the estimation.

The primary measure is the Score, which is designed to treat \( FN \) with a much higher importance. The AUC measure is provided for informational purposes only.

As shown in Figure 4, three types of arrhythmia—i.e., ventricular flutter or fibrillation, ventricular tachycardia, and asystole—are very difficult to predict, especially in light of the results for the validation dataset. Nevertheless, application of the proposed BSS ANN delivered an improvement in terms of classification accuracy. Importantly, the accuracy of the BSS ANN 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. Specifically, the following scores were observed for the BSS ANN:

- 0.632 for ventricular tachycardia—accuracy of the model was improved in comparison to the ANN (0.540), RF (0.315), and RPART (0.375) methods;
- 0.804 for ventricular fibrillation—accuracy of the model was improved in comparison to the ANN (0.691), RF (0.306), and RPART (0.414) methods;
- 0.907 for tachycardia—accuracy of the model was improved in comparison to the ANN (0.903), RF (0.872), and RPART (0.717) methods;
- 0.756 for bradycardia—accuracy of the model was improved in comparison to the RPART method (0.706), whereas the ANN (0.775) and RF (0.757) methods delivered better accuracy;
- 0.708 for asystole—accuracy of the model was improved in comparison to the ANN (0.688), RF (0.618), and RPART (0.640) methods.

As far as BSS aggregation applied to an ANN is concerned, the results indicate that it is a viable approach leading to the improvement of classification, especially when base methods, such as the ANN and RF/decision tree methods are not able to deliver acceptable accuracy.
Figure 4. AUC and score results for each type of arrhythmia.
5. Conclusions

The problem of false arrhythmia alarms in ICUs is still a demanding task for the algorithms, as there could be a number of potential triggers for false alarms, including noises and device malfunctions that highly influence the analyzed signals and, thus, the model performance.

The novelty of this research is the demonstration of a neural network enhanced by a BSS design. It is robust and can deal with arrhythmia types that are difficult to predict correctly, i.e., ventricular tachycardia, ventricular flutter/fibrillation, and asystole, particularly in noisy signals. The results proved that BSS ANNs are able to detect four arrhythmias—i.e., ventricular tachycardia, ventricular fibrillation, tachycardia, and asystole—with better accuracy than the benchmarking models. For bradycardia, the results of BSS ANNs are slightly worse than those observed for the ANN or RF. However, it is important to acknowledge that bradycardia and tachycardia are relatively easy to predict, and current algorithms already present high performance.

Importantly, we showed that the proposed approach of BSS aggregation is highly competitive compared to other machine learning methods.

We believe that reducing the number of false alarms and avoiding the suppression of true ones are important goals; therefore, the study can be further enhanced by including additional ICA algorithms in the design, and looking for higher classification accuracy. As a result this may lead to the analysis of the algorithms’ structure and diversity and their effects on classification.

Author Contributions: K.G. prepared the simulation and analysis and wrote Sections 1–5 of the manuscript; I.G. wrote Sections 1 and 3 of the manuscript; M.G. wrote Section 2 of the manuscript; T.Z. wrote Sections 1, 4 and 5 of the manuscript; all authors read and approved the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Pan, J.; Tompkins, W.J. A Real-Time QRS Detection Algorithm. IEEE Trans. Biomed. Eng. 1985, 32, 230–236. [CrossRef]
2. 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. [CrossRef] [PubMed]
3. Silva, I.; Moody, B.; Behar, J.; Johnson, A.; Clifford, G.D.; Moody, G.B. Robust detection of heart beats in multimodal data. Physiol. Meas. 2015, 36, 1629–1644. [CrossRef] [PubMed]
4. Behar, J.; Johnson, A.; Clifford, G.D.; Oster, J. A comparison of single channel fetal ECG extraction methods. Ann. Biomed. Eng. 2014, 42, 1340–1353. [CrossRef] [PubMed]
5. Clifford, G.D.; Silva, I.; Moody, B.; Li, Q.; Kella, D.; Shahin, A.; Kooistra, T.L.; Perry, D.; Mark, R.G. The PhysioNet/competing 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. [CrossRef]
6. 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. [CrossRef] [PubMed]
7. Rooijakkers, M.J.; Rabotti, C.; Oei, S.G.; Mischi, M. Low-complexity R-peak detection for ambulatory fetal monitoring. Physiol. Meas. 2012, 33, 1135–1150. [CrossRef] [PubMed]
8. Gieraltowski, J.; Ciuchciński, K.; Grzegorczyk, J.; Kośna, K.; Soliński, M.; Podziemski, P. RS slope detection algorithm for extraction of heart rate from noisy, multimodal recordings. Physiol. Meas. 2015, 36, 1743–1761. [CrossRef]
9. 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-implementationecg-qrs-detector (accessed on 10 July 2019).
10. Antink, C.H.; Leonhardt, S.; Walter, M. Reducing false alarms in the ICU by quantifying self-similarity of multimodal biosignals. *Physiol. Meas.* 2016, 37, 1233–1252. [CrossRef]

11. Kalidas, V.; Tamil, L.S. Cardiac arrhythmia classification using multi-modal signal analysis. *Physiological measurement* 2016, 37, 1233–1252. [CrossRef]

12. Sadr, N.; Huvanandana, J.; Nguyen, D.T.; Kalra, C.; McEwan, A.; de Chazal, P. Reducing false arrhythmia alarms in the ICU using multimodal signals and robust QRS detection. *Physiological measurement* 2016, 37, 1340. [CrossRef]

13. 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. [CrossRef] [PubMed]

14. Khadra, L.; Al-Fahoum, A.S.; Al-Nashash, H. Detection of life-threatening cardiac arrhythmias using the wavelet transformation. *Med. Biol. Eng. Comput.* 1997, 35, 626–632. [CrossRef] [PubMed]

15. Christov, I.I. Real time electrocardiogram QRS detection using combined adaptive threshold. *Biomed. Eng. Online* 2004, 3, 28. [CrossRef] [PubMed]

16. Arzeno, N.M.; Deng, Z.-D.; Poon, C.-S. Analysis of First-Derivative Based QRS Detection Algorithms. *IEEE Trans. Biomed. Eng.* 2008, 55, 478–484. [CrossRef]

17. Mjahad, A.; Rosado-Muñoz, A.; Bataller-Mompeán, M.; Francès-Villora, J.V.; Guerrero-Martínez, J.F. 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. [CrossRef]

18. Chen, S.; Thakor, N.V.; Mower, M.M. Ventricular fibrillation detection by a regression test on the autocorrelation function. *Med. Biol. Eng. Comput.* 1987, 25, 241–249. [CrossRef]

19. Balasundaram, K.; Masse, S.; Nair, K.; Farid, T.; Nanthakumar, K.; Umaphaty, 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. [CrossRef]

20. Li, H.; Han, W.; Hu, C.; Meng, M.Q.-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. [CrossRef]

21. Alonso-Atienza, F.; Morgado, E.; Fernandez-Martinez, L.; Garcia-Alberola, A.; Rojo-Alvarez, J.L. Detection of Life-Threatening Arrhythmias Using Feature Selection and Support Vector Machines. *IEEE Trans. Biomed. Eng.* 2014, 61, 832–840. [CrossRef]

22. Asadi, F.; Mollakazemi, M.J.; Ghiasi, S.; Sadati, S.H. 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. [CrossRef]

23. Eerikäinen, L.M.; Vanschoren, J.; Rooijakkers, M.J.; Vullings, R.; Aarts, R.M. Reduction of false arrhythmia alarms using signal selection and machine learning. *Physiol. Meas.* 2016, 37, 1204–1216. [CrossRef]

24. Luo, Z.; Li, C.; Zhu, L. A Comprehensive Survey on Blind Source Separation for Wireless Adaptive Processing: Principles, Perspectives, Challenges and New Research Directions. *IEEE Access* 2018, 6, 66685–66708. [CrossRef]

25. Klaus, N.; Hannu, O. Independent component analysis: A statistical perspective. *Wiley Interdiscip. Rev. Comput. Stat.* 2018, 10, e1440. [CrossRef]

26. Bobin, J.; Rapin, J.; Larue, A.; Starck, J.L. Sparsity and adaptivity for the blind separation of partially correlated sources. *IEEE Trans. Signal Process.* 2015, 63, 1199–1213. [CrossRef]
31. Cichocki, A.; Zdunek, R.; Phan, A.H.; Amari, S.I. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation; John Wiley & Sons: Hoboken, NJ, USA, 2009.
32. Cruces, S. Bounded component analysis of noisy underdetermined and overdetermined mixtures. IEEE Trans. Signal Process. 2015, 63, 2279–2294. [CrossRef]
33. Sagi, O.; Rokach, L. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2018, 8, e1249. [CrossRef]
34. Ren, Y.; Zhang, L.; Suganthan, P.N. Ensemble Classification and Regression—Recent Developments, Applications and Future Directions. IEEE Comput. Intell. Mag. 2016, 11, 41–53. [CrossRef]
35. Breiman, L. Bagging predictors. Mach. Learn. 1996, 24, 123–140. [CrossRef]
36. Breiman, L. Random forests. Mach. Learn. 2001, 45, 261–277. [CrossRef]
37. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the ICML’96 Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
38. Friedman, J.H. Stochastic gradient boosting. Comput. Stat. Data Anal. 2002, 38, 367–378. [CrossRef]
39. Gajowniczek, K.; Karpio, K.; Łukasiewicz, P.; Orlowski, A.; Żabkowski, T. Q-Entropy Approach to Selecting High Income Households. Acta Phys. Pol. A 2015, 127, A-38–A-44. [CrossRef]
40. Gajowniczek, K.; Orlowski, A.; Żabkowski, T. Simulation Study on the Application of the Generalized Entropy Concept in Artificial Neural Networks. Entropy 2018, 20, 249. [CrossRef]
41. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. Neurocomputing 2017, 234, 11–26. [CrossRef]
42. Szupiluk, R.; Wojewnik, P.; Żabkowski, T. Prediction Improvement via Smooth Component Analysis and Neural Network Mixing. Lect. Notes Comput. Sci. 2006, 133–140. [CrossRef]
43. Wright, M.N.; Ziegler, A. Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. J. Stat. Softw. 2017, 77. [CrossRef]
44. © 2020 by the authors. 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/).