

# Automated Atrial Fibrillation Source Detection Using Shallow Convolutional Neural Networks

Isac N. Lira<sup>1</sup>, Pedro Marinho R. de Oliveira<sup>2</sup>, Walter Freitas Jr.<sup>1</sup>, Vicente Zarzoso<sup>2</sup>

<sup>1</sup> Université Côte d'Azur, CNRS, I3S Laboratory, Sophia Antipolis, France

<sup>2</sup> Universidade Federal do Ceará, Brazil

## Abstract

*Atrial fibrillation (AF) is the most frequent sustained arrhythmia diagnosed in clinical practice. Understanding its electrophysiological mechanisms requires a precise analysis of the atrial activity (AA) signal in ECG recordings. Over the years, signal processing methods have helped cardiologists in this task by noninvasively extracting the AA from the ECG, which can be carried out using blind source separation (BSS) methods. However, the robust automated selection of the AA source among the other sources is still an open issue. Recently, deep learning architectures like convolutional neural networks (CNNs) have gained attention mainly by their power of automatically extracting complex features from signals and classifying them. In this scenario, the present work proposes a shallow CNN model to detect AA sources with an automated feature extraction step overcoming the performance of other methods present in the literature.*

## 1. Introduction

The most frequent sustained arrhythmia diagnosed in clinical practice, atrial fibrillation (AF) is a supraventricular tachyarrhythmia characterized by an uncoordinated and irregular atrial activation [1]. The electrophysiological mechanisms underlying AF are not totally understood, which makes this cardiac condition gather increasing attention from researchers and cardiologists in the past few years.

Signal processing methods have helped cardiologists to better manage this cardiac rhythm disturbance by noninvasively extracting the atrial activity (AA) signal from the standard 12-lead electrocardiogram (ECG). In particular, the AA extraction from multilead AF ECGs accepts a blind source separation (BSS) formulation [2] and several techniques to solve BSS problems were reported in the literature as useful tools in noninvasive AA extraction for AF analysis [2]–[5]. In particular, the block term decomposition (BTD), a tensor-based BSS method, have proved to be

an important AA extraction tool, overcoming the matrix-based techniques [4], [5].

In the challenging case of AF ECGs, techniques to solve BSS problems separate the original recording in several sources, where, typically, at least one of these sources contains the AA. After separating the source signals, it is necessary to select the AA source estimate among the other sources. Atrial source selection requires visual inspection to achieve optimality, as an optimal automated method for this task is still an open challenge. However, machine learning algorithms have provided an improved accuracy compared to other automated methods [7]. In this context, deep learning techniques have proved to be very efficient in many tasks like image detection and classification, mainly due to its ability to perform an automated feature extraction [8].

In the previous work [7] standard machine learning models were applied to detect AA sources using hand-crafted features. In order to improve this process, the present work proposes a framework that combines the BTD technique with a shallow convolutional neural network (CNN) in order to tackle the task of detecting AA sources from ECG segments with automatically extracted features.

## 2. Data Description and Pre-processing

### 2.1. Database

The analyzed data during the experiments are composed by 116 random segments of 58 12-lead ECG recordings from 58 persistent patients, each one with duration of approximately 1 second. These recordings belong to a database provided by the Cardiology Department of Princess Grace Hospital Center, Monaco. The recordings are acquired at a 977 Hz sampling rate and are pre-processed by a zero-phase forward-backward type-II Chebyshev bandpass filter with cutoff frequencies of 0.5 and 40 Hz, in order to suppress high-frequency noise and baseline wandering.

## 2.2. AA Source Extraction

Signal processing techniques that solve BSS problems separate the observed ECG signal matrix  $\mathbf{Y}$  in a linear combination of a mixing matrix  $\mathbf{M}$  and a source matrix  $\mathbf{S}$ :

$$\mathbf{Y} = \mathbf{M}\mathbf{S} \in \mathbb{R}^{\mathbb{K} \times \mathbb{N}} \quad (1)$$

In the present case of study,  $\mathbf{Y} \in \mathbb{R}^{\mathbb{K} \times \mathbb{N}}$  is the AF ECG data matrix, composed of  $K$  signals (leads) and  $N$  samples,  $\mathbf{M} \in \mathbb{R}^{\mathbb{K} \times \mathbb{R}}$  is the mixing matrix, modeling the propagation of the  $R$  cardiac electrical sources from the heart to the  $K$  leads in the body surface, and  $\mathbf{S} \in \mathbb{R}^{\mathbb{R} \times \mathbb{N}}$  is the source matrix that contains  $R$  sources, mainly atrial, ventricular and noise sources.

After some mathematical manipulations, the ECG data matrix can be transformed into a tensor that admits a BTD model [5]. This tensor factorization technique is based on a third-order tensor  $\mathcal{Y}$  built from Hankel matrices that are constructed from each row of the observed data matrix. The tensor is then decomposed as [6]:

$$\mathcal{Y} = \sum_{r=1}^R \mathbf{E}_r \circ \mathbf{c}_r \quad (2)$$

where  $\circ$  represents the outer product,  $\mathbf{c}_r$  is a nonzero vector,  $\mathbf{E}_r$  is a Hankel matrix built from each source and  $R$  is the number of sources.

This tensor-based BSS technique is computed using the recently proposed algorithm called constrained alternating group lasso [4] and applied to ECG segments generating 509 sources that are visually labeled as 122 AA sources, 273 ventricular activity (VA) sources and 114 unknown (UNK) sources.

The scheme proposed in this work aims to distinguish only AA sources from the remaining sources. In this way, the VA and UNK sources are grouped into a single class, called non-AA sources, which configures a binary classification problem to be solved.

Initially, the data is randomly split into train and validation sets with a ratio of 80% for the training samples. The signals are normalized with respect to the mean  $\mu_t$  and variance  $\sigma_t$  obtained from the training set  $\mathbf{X}_{\text{train}} = \{\mathbf{x}_0, \dots, \mathbf{x}_T\}$ , where  $\mathbf{x}_i$  is a vector representing an ECG source signal. Considering  $\underline{\mathbf{v}}$  the concatenation of all vectors from  $\mathbf{X}_{\text{train}}$ , one can compute the overall  $\mu_t$  and  $\sigma_t$  as:

$$\mu_t = \frac{1}{V} \sum_{i=1}^V v_i \quad (3a)$$

$$\sigma_t = \frac{1}{V} \sum_{i=1}^V (v_i - \mu_t)^2 \quad (3b)$$

where  $v_i$  is the  $i$ -th entry of  $\underline{\mathbf{v}}$ , and  $V = |\underline{\mathbf{v}}|$  denotes the cardinality of  $\underline{\mathbf{v}}$ . The transformed signal components are then computed as:

$$z_i = \frac{x_i - \mu_t}{\sigma_t}, \text{ for } i = 1, \dots, |\underline{\mathbf{x}}|. \quad (4)$$

This normalization operation is assigned to a batch normalization layer in the CNN model.

## 2.3. Data Augmentation

Due to the low number of segments available, the training of deep learning models can suffer from overfitting. To overcome this problem, a window slicing (WS) based method is applied to augment the data and consequently provides more samples to the training process. This method was first introduced in [11] also in the context of time series classification using CNN and it has proved useful to increase model performance. It affects the training as well as the prediction phase.

For a given ECG segment and its class  $(\underline{\mathbf{x}}_i, y_i)$ , a window with size  $W < |\underline{\mathbf{x}}_i|$  is applied to extract a subsignal  $\underline{\mathbf{x}}_{i,j}$ . The window is moved by  $S \leq W$  samples to obtain a new signal  $\underline{\mathbf{x}}_{i,j+1}$  and the process is repeated until the original segment is completely split.

By applying the WS strategy during the training step, each signal  $\underline{\mathbf{x}}_i$  generates a set of subsignals  $X_i = \{(\underline{\mathbf{x}}_{i,0}, y_i), \dots, (\underline{\mathbf{x}}_{i,N}, y_i)\}$ , all of them sharing the same label  $y_i$ . For the prediction phase, we propose to estimate the value of the class probability  $\tilde{y}_i$  by averaging the model scores of the subsignals, as described in Equation (5).

$$\tilde{y}_i = \frac{1}{N} \sum_{j=0}^N \tilde{y}_{i,j} \quad (5)$$

where  $N$  is the number of the generated subsignals from  $\underline{\mathbf{x}}_i$  and  $\tilde{y}_{i,j}$  the model prediction for  $\underline{\mathbf{x}}_{i,j}$ . Just after the data augmentation process, a random oversampling (ROS) step is applied over the minority class (AA source) in order to balance the training data.

## 3. CNN Model Selection

### 3.1. Convolutional Neural Networks

The CNN is a Deep Learning model initially designed for multi-dimensional data like images. The main components of a CNN are the convolutional layers, the pooling layers and fully connected layers. During the convolutional operation, a bank of filters is applied over the whole input signal using the same weights and it generates activations for each receptive field that are combined to form a feature map [10]. Each set of weights are optimized by a

gradient algorithm to detect specific type of features along the input signal.

Along with the convolutional operation, the pooling layers perform a reduction in the feature space and combine similar features [14]. For example, the max-pooling kernel slides the feature space getting the maximum value from small regions.

### 3.2. Architecture Optimization

To find a suitable CNN configuration for the task of AA source detection, a Bayesian algorithm is applied using BoTorch [13], a framework used for optimization tasks. We compare different shallow CNN architectures changing the following parameters: number of hidden nodes, training epochs, convolutional/pooling layers, kernel size, kernel stride and batch size.

The maximal number of convolutional layers is set to 3 which keeps the CNN model simpler and much shallower than the common CNN models found in the literature. This reduces the number of trainable weights, thus avoiding overfitting. Along with the model architecture parameters, the augmentation settings (window size and stride) are also optimized. Furthermore, the upper bound value for the window size is limited by the length of the shortest extracted ECG source.

Let  $L$  be the number of convolutional layers and  $k_{i,l}$  the size of the convolutional kernel  $i$  in the layer  $l$ . The constraint  $k_{i,l} \leq k_{i,l+1}$  is imposed on each layer  $l \in \{1, \dots, L - 1\}$ . Another constraint requires all kernels from layer  $l$  to have the same size  $K_l$ .

Similarly, the stride  $S_l$  for the kernels have to follow the inequality  $S_l \leq S_{l+1}$ . By doing that, it is produced an increasing reduction in the feature space.

A final constraint is defined to have an increasing number of channels in consecutive layers which allows the model to capture more complex features from the signals.

### 3.3. Model Training and Evaluation

The weights for the shallow CNN models are optimized using the Adam optimizer with a learning rate being selected by the Bayesian algorithm. Their values are within the range  $[10^{-4}, 10^{-3}]$ . Each model is evaluated on the validation set with respect to the Area Under the ROC Curve (AUC) aiming to find the model that provides the maximum possible score.

In this work, we consider the AA sources as being the positive class, and the non-AA sources as the negative one. The sensitivity and specificity metrics are used to measure the model performance for each class individually. The sensitivity is defined as:

$$SEN = \frac{TP}{TP + FN} \quad (6)$$

where  $TP$  are the true positive samples and  $FN$  the False Negatives. Similarly, the specificity is computed by (7) using the  $TN$  as the number of true negative samples and  $FP$  the quantity of false positive samples:

$$SPE = \frac{TN}{TN + FP}. \quad (7)$$

Finally, the accuracy is computed to measure the overall model precision.

## 4. Experimental Results

After running 100 trials, the best architecture is chosen with an AUC validation score of 97.5%. The best parameters for the CNN are in the Table 1. The most appropriate batch size shown is 124 samples, and for the augmentation window size the best value is 472 with a stride percentage of 21% resulting in an absolute stride of 99 samples.

Table 1. Optimized parameters for the shallow CNN architecture.

Layer	Kernel Size	Strides	Output Size
ECG Signal	-	-	$1 \times 472$
Batch Normalization	-	-	$1 \times 472$
Convolution	$1 \times 4$	3	$19 \times 157$
ReLU	-	-	$19 \times 157$
MaxPool	$1 \times 2$	1	$19 \times 156$
Convolution	$1 \times 8$	3	$29 \times 50$
ReLU	-	-	$29 \times 50$
MaxPool	$1 \times 10$	3	$29 \times 14$
Dropout	-	-	$29 \times 14$
Linear	-	-	$1 \times 762$
Dropout	-	-	$1 \times 762$
Linear	-	-	$1 \times 2$
Softmax	-	-	$1 \times 2$

The model evaluation is performed applying a 10-fold cross validation (CV) to compute ACC and AUC. Not all data are used in the evaluation; instead, the CV folds are computed only over the training data, since the CNN architecture is selected on the validation set. The average AUC and ACC achieved are 96.3% and 93.6%, respectively and the results across the CV folds are plotted in a box plot in Figure 1.

Additionally the model performance is represented in a confusion matrix in Figure 2 whose values are based on the CV folds. From the matrix, the obtained sensitivity and specificity metrics are 91.75% and 94.19%, respectively.

## 5. Conclusions

In order to improve the process of detecting AA sources in ECG segments, the present work proposes a framework

to perform an automated source detection. The solution joins a tensor-based BSS method with a shallow CNN model with an optimized structure. The model requires less training weights than common deep learning models and it also has the advantage of automatically extracting features from ECG sources, thus avoiding handcrafted features and achieving promising results.

Further works may include the application of transfer learning using ECG source spectrograms to increase the detection performance.

## Acknowledgments

Pedro Marinho R. de Oliveira is funded by a PhD scholarship from the IT Doctoral School (EDSTIC) of the Université Côte d’Azur.

## References

- [1] C. T. January, L. S. Wann, H. Calkins, L. Y. Chen, J. E. Cigarroa, et al., “2019 AHA/ACC/HRS Focused Update of the 2014 AHA/ACC/HRS Guideline for the Management of Patients With Atrial Fibrillation”, *Journal of the American College of Cardiology*, vol. 74, no. 1, pp. 104- 132, 2019.
- [2] J. J. Rieta, F. Castells, C. Sánchez, V. Zarzoso, and J. Millet, “Atrial activity extraction for atrial fibrillation analysis using blind source separation”, *IEEE Trans. Biomed. Eng.*, vol. 51, no. 7, pp. 1176-1186, Jul. 2004.
- [3] F. Castells, J. J. Rieta, J. Millet, and V. Zarzoso, “Spatiotemporal blind source separation approach to atrial activity estimation in atrial tachyarrhythmias”, *IEEE Trans. Biomed. Eng.*, vol. 52, no. 2, pp. 258-267, Feb. 2005.
- [4] J. H. de M. Goulart, P. M. R. de Oliveira, R. C. Farias, V. Zarzoso, and P. Comon “Alternating group lasso for block-term tensor decomposition with application to ECG source separation”. *IEEE Trans. Sig. Proc.*, vol. 68, pp. 2682–2696, 2020.
- [5] P. M. R. de Oliveira and V. Zarzoso, “Block term decomposition of ECG recordings for atrial fibrillation analysis: Temporal and inter-patient variability”, *Journal of Comm. Inf. Syst.*, vol. 34, no. 1, pp. 111-119, 2019.
- [6] L. De Lathauwer, “Blind separation of exponential polynomials and the decomposition of a tensor in rank- $(l_r, l_r, 1)$  terms”, *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1451–1474, 2011.
- [7] P. M. R. de Oliveira, V. Zarzoso and C.A.R. Fernandes, “Source classification in atrial fibrillation using a machine learning approach”, in *Proc. CinC-2019, 46th Computing in Cardiology Conference*, Biopolis, Singapore, Sep. 8-11, pp. 1-4, 2019.
- [8] Litjens, Geert, et al. “A survey on deep learning in medical image analysis.” *Medical image analysis* 42 (2017): 60-88.
- [9] Le Guennec, Arthur, Simon Malinowski, and Romain Tavenard. “Data augmentation for time series classification using convolutional neural networks.” 2016.
- [10] Johnson, Justin M., and Taghi M. Khoshgoftaar. “Survey on deep learning with class imbalance.” *Journal of Big Data* 6.1 (2019): 27.

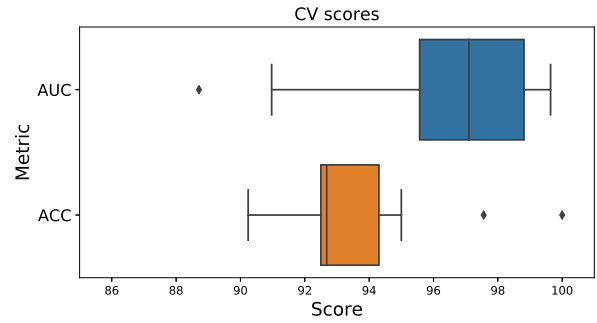


Figure 1. Box plot of the AUC and ACC metrics over CV.

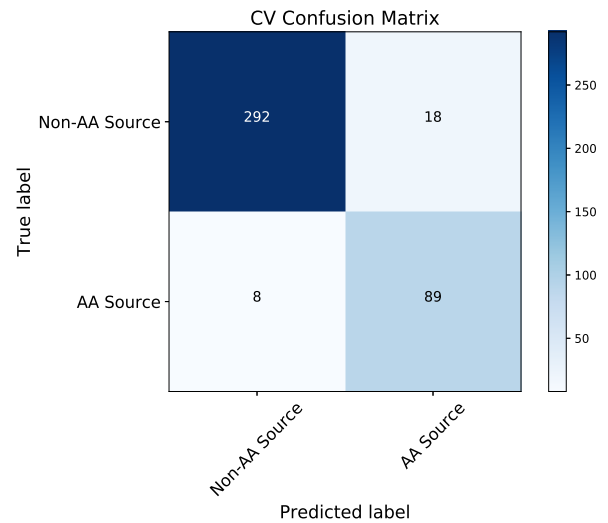


Figure 2. CV confusion matrix.

- [11] Cui, Zhicheng, Wenlin Chen, and Yixin Chen. “Multi-scale convolutional neural networks for time series classification.” arXiv preprint arXiv:1603.06995 (2016).
- [12] Van Hulse, Jason, Taghi M. Khoshgoftaar, and Amri Napolitano. “Experimental perspectives on learning from imbalanced data.” *Proceedings of the 24th international conference on Machine learning*. 2007.
- [13] Balandat, Maximilian, et al. “Botorch: Programmable bayesian optimization in pytorch.” arXiv preprint arXiv:1910.06403 (2019).
- [14] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” *nature* 521.7553 (2015): 436-444.

Address for correspondence:

Pedro Marinho R. de Oliveira  
 Université Côte d’Azur, CNRS, I3S Laboratory  
 06903 Sophia Antipolis Cedex  
 marinho@i3s.unice.fr