Automated Atrial Fibrillation Source Detection using Shallow Convolutional Neural Networks

Isac N. Lira[1], Pedro Marinho R. de Oliveira[2], Walter Freitas Jr.[1] Vicente Zarzoso[2]
[1]Universidade Federal do Ceará, Brazil
[2]Université Côte d'Azur, CNRS, I3S Laboratory, France

***Abstract*—Atrial fibrillation (AF) is the most frequent sustained arrhythmia diagnosed in clinical practice. Understanding its ectrophysiological 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 extract complex features from signals and classifying them. In this scenario, the present work proposes a shallow CNN model to automatically detect AA sources without the need of hands-on feature extraction steps. The parameters for the model architecture (i.e, kernel size) are selected by a Bayesian optimization algorithm, and the final model achieved promising results overcoming the performance of all the methods present in literature.**

I. Introduction

Being 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 increasingly attention of 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). Indeed, 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 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 mainly contains the AA. After separating the source signals, it is necessary to select the AA source estimate among the other sources. For this task, machine learning algorithms have provided an improved accuracy compared to other automated methods [8]. Atrial source selection requires visual inspection to achieve optimality, as an optimal automated method for this task is still an open challenge. 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 [9].

In this work a framework that combines a tensor-based technique with a shallow convolutional neural network (CNN) is designed to tackle the task of detecting AA sources from ECG segments without handcrafted features.

II. Methodology

A. 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 $\mathbf{S}$:

$$\mathbf{Y} = \mathbf{MS} \in \mathbb{R}^{K \times N} . \tag{1}$$

In the present case of study, $\mathbf{Y} \in \mathbb{R}^{K \times N}$ is the AF ECG data matrix, composed of K signals (leads) and N samples, $\mathbf{M} \in \mathbb{R}^{K \times R}$ 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 $\mathbf{S} \in \mathbb{R}^{R \times N}$ 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 block term decomposition (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]:

$$\mathcal{Y} = \sum_{r=1}^{R} \mathbf{E}_r \circ \mathbf{c}_{\cdot r} \tag{2}$$

where $\circ$ represents the outer product, $\mathbf{c}_{\cdot r}$ is a nonzero vector, $\mathbf{E}_r$ is a Hankel matrix built from each source and R is the number of sources. This technique suits the characteristics of AA during AF, since atrial signals can be approximated by all-pole models and mapped onto Hankel matrices with rank equal to the number of poles [7].

This tensor-based BSS technique is computed using the recently proposed algorithm called constrained alternating group lasso [4] and applied to 116 random segments of 58 12-lead ECG recordings from 58 persistent patients, generating 509 sources that are visually labeled as 122 AA sources, 273 ventricular activity (VA) sources and 114 unknown (UNK) sources, 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 preprocessed 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.

Regardless the number of groups, the proposed schema in this work aims to distinguish only AA sources from the remaining sources. In this way, the VA and UNK sources are enclosed into a single class, the non-AA source, which configures a binary classification problem to be solved.

Initially, the data is randomly splitted into train and validation sets. The signals were normalized with respect to the mean μ_t and variance σ_t obtained from the training matrix $\mathbf{X_{train}} = [\mathbf{x_0}, \ldots, \mathbf{x_T}]$, where $\mathbf{x_i}$ is a vector representing an ECG source signal in the training set. Considering $\mathbf{v} = \text{vec}(\mathbf{X_{train}})$, where vec is the operation for matrix vectorization, one can compute μ_t and σ_t as:

$$\mu_t = \frac{1}{V}\sum_{i=1}^{V} v_i \tag{3a}$$

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

where $V = |\mathbf{v}|$ denotes the cardinality of $\mathbf{v}$. The transformed signals are then computed as:

$$z_i = \frac{x_i - \mu_t}{\sigma_t}, \text{ for } i = 1, \ldots, X. \tag{4}$$

This normalization operation was assigned to a batch normalization layer in the CNN model as will be explained in coming sessions.

B. Oversampling

The distribution for the AA class is about 24% of the extracted sources. It leads to a class imbalance problem that can force the model to learn mostly from the majority class due to its higher prior probability [11]. However, the challenge of dealing with imbalanced data is well-posed and it already has robust and straightforward solutions that enable machine learning models to learn from both classes in a more balanced way [13].

In this context, the random oversampling (ROS) is a strategy that duplicates random samples from the minority class in the training set which can alleviate the model bias towards the majority class.

C. Data Augmentation

Due to the low number of segments available, the training of Deep Learning models can suffer from overfitting. To overcame 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 [12] also in the context of time series classification using CNN and it has been proved to be useful to increase the model performance. It affects the training as well as the prediction phase.

For a given ECG segment and its class $(\mathbf{x}_i, y_i)$, a window with size $W < |\mathbf{x}_i|$ is applied to extract a subsignal $\mathbf{x}_{i,j}$. The window is moved by $S \leq W$ samples to obtain a new signal $\mathbf{x}_{i,j+1}$ and the process is repeated until the original segment is completely splitted. The augmentation parameter S is the stride, and similarly to the window size parameter it can be selected such that the model performance is improved.

By applying the WS strategy during the training, each signal $\mathbf{x}_i$ generates a set of subsignals $\mathcal{X}_i = \{(\mathbf{x}_{i0}, y_i), \ldots, (\mathbf{x}_{iN}, y_i)\}$, all of them sharing the same label y_i. The figure 1 illustrates the split process that divide the input segment in three subsignals. 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 bellow in the equation 5.

$$\tilde{y}_i = \frac{1}{N}\sum_{j=0}^{N} \tilde{y}_{ij} \tag{5}$$

where N is the number of the generated subsignals from $\mathbf{x}_i$ and $\tilde{y}_{ij}$ the model prediction for $\mathbf{x}_{ij}$.

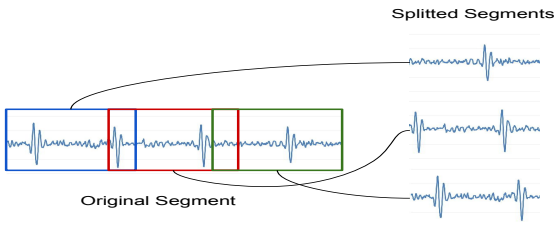

Fig. 1. An ilustration of the augmentation process with 3 steps.

III. Model Selection

A. 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 layer. During the convolutional operation, a bank of filters are 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 [11]. 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 [15]. For example, the max-pooling kernel slides the feature space getting the maximum value from small regions.

B. Architecture Optimization

To find a suitable CNN configuration for the task of AA source detection, a Bayesian algorithm is applied using the BoTorch [14] comparing different shallow CNN architectures based on the following search space:

- Number of Hidden Nodes: $\{10, \ldots, 1000\}$
- Number of Training Epochs: $\{10, \ldots, 200\}$
- Number of Convolutional/Pooling layers: $\{1, \ldots, 3\}$
- Convolutional Kernel Size: $\{2, \ldots, 36\}$

- Convolutional Stride: $\{1, \ldots, 6\}$
- Max-pooling Kernel Size: $\{2, \ldots, 15\}$
- Max-pooling Stride: $\{1, \ldots, 3\}$
- Batch Size: $\{20, \ldots, 200\}$

The maximal number of convolutional layer is set to 3 which keeps the CNN model simpler and much shorter than the common CNN models found in the literature. This reduces the number of trainable weights avoiding overfitting.

Besides the optimization parameters listed above, the augmentation parameters are also optimized and their range are described below:

- Window Size (W): $\{64, \ldots, 512\}$
- Stride (S) : $\{0.01, \ldots, 0.75\}$

where the stride is here represented as a percentage of the window size to simplify the search and avoid constraints during the optimization process. Furthermore, the upper bound value for the window size is limited by the length of the shortest extracted ECG source.

Be L 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 $l \in \{1, \ldots, L-1\}$. 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, we produce an increasing reduction in the feature space since the layers output is defined by the equation (6) below.

$$O_l = \lfloor \frac{N_l - K_l - 2}{S_l} + 1 \rfloor, \tag{6}$$

where N_l is the size of the input signal, and O_l is the output size for the layer l. The symbol $\lfloor$ stands for the floor operation. The higher is the stride S_l the lower is the output size O_l.

A final constraint is defined to have an increasing number of channels in the layers which allows the model to capture more complex signal partners.

C. 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 $[10^{-4}, 10^{-3}]$. The data is splitted in train and validation sets with a ratio of 80/20, respectively. 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 score possible.

In this work, we consider the AA sources as being the positive class, and the non-AA sources the negative one. The sensitivity and specificity metrics are used to measure the model performance for each class individually. The sensitivity is defined by the equation 7.

$$sensitivity = \frac{TP}{TP + FN} \tag{7}$$

where TP are the True Positive samples and FN the False Negatives. Similarly, the specificity is computed by the equation 8 using the TN as the number of True Negative samples and FP the quantity of False Positive samples.

$$specificity = \frac{TN}{TN + FP} \tag{8}$$

Finally, the accuracy (ACC) is computed to measure the overall model correctness, and it is defined by the equation 9.

$$ACC = \frac{TN + TP}{TN + FN + TP + FP} \tag{9}$$

IV. Experiments and Results

After run 100 trials, the best architecture was chosen with an AUC validation score of 97.5. The best parameters for the CNN are showed in the table I. The more appropriate batch size was 124 samples, and for the augmentation window size the best value was 472 with a stride percentage of 21% resulting in a absolute stride of 99 samples.

TABLE I
Optimized parameters for the shallow CNN architecture.

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

The model evaluation is performed applying a 10-fold cross validation (CV) to compute accuracy and AUC. Not all data is used in the evaluation, instead the scores are computed only over the training data since the CNN architecture was selected on the validation set. The obtained AUC and accuracy scores stratified by fold is showed in the table II.

TABLE II
Computed metrics on CV folds

Metric/Fold	Cross Validation Metrics										
	K1	*k2*	*k3*	*k4*	*K5*	*k6*	*k7*	*k8*	*k9*	*k10*	*Mean*
AUC	98.39	98.71	99.35	88.71	90.97	97.1	94.84	99.64	96.42	98.92	96.3
ACC	92.68	100.0	97.56	92.68	92.68	90.24	90.24	95.0	92.5	92.5	93.61

These CV results are plotted in a box plot in the graph 2 below. The average AUC and ACC achieved are 96.3% and 93.6%, respectively.

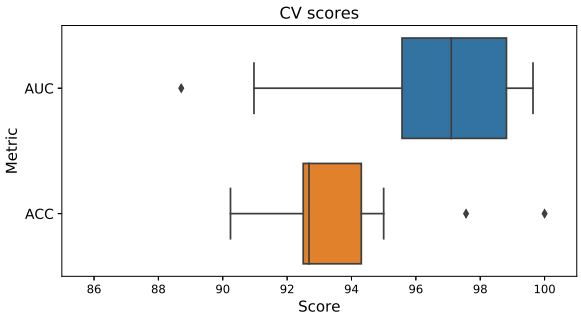

Fig. 2. Box plot with the AUC and ACC metrics over CV

Additionally the model performance is represented in a confusion matrix in the figure 3 whose values are based on the CV folds. From the matrix, the obtained sensitivity and specificity metrics are 91.75% and 94.19%, in that order.

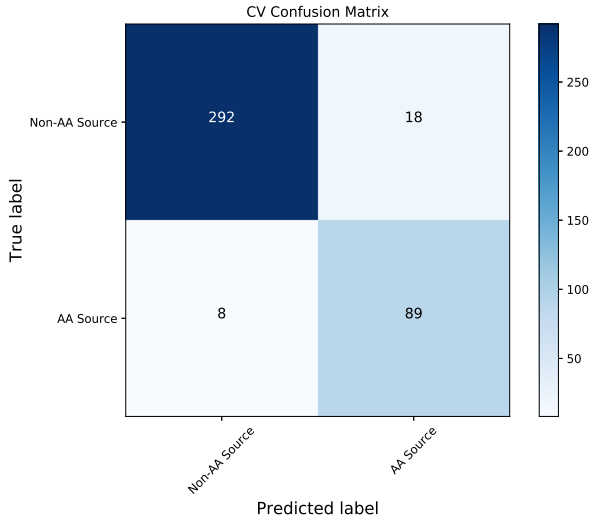

Fig. 3. CV Confusion Matrix

REFERENCES

[1] C. T. January, L. S. Wann, J. S. Alpert, H. Calkins, J. C. Cleveland, J. E. Cigarroa, J. B. Conti, *et al.*, "2014 AHA/ACC/HRS guideline for the management of patients with atrial fibrillation: A report of the American College of Cardiology/American Heart Association Task Force on practice guidelines and the Heart Rhythm Society", *Circulation*, vol. 64, no. 21, pp. 2246-2280, Dec. 2014.

V. CONCLUSION

In this work we propose a framework to deal with the automated detection of atrial fibrillation (AA) sources from ECG segments. The proposed solution joins a tensor-based BSS method with a shallow CNN model, achieving promising results in automatically detecting AA sources among the others, after BSS.

[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 Transactions on Biomedical Engineering*, 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 Transactions on Biomedical Engineering*, 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 Transactions on Signal Processing*, 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 Communication and Information Systems*, 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] V. Zarzoso, "Parameter estimation in block term decomposition for noninvasive atrial fibrillation analysis", in *Proc. CAMSAP-2017, IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Curaçao, Dutch Antilles, pp. 1–5, Dec. 10-13, 2017.

[8] 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.

[9] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[10] Le Guennec, A., Malinowski, S., Tavenard, R. (2016, September). Data augmentation for time series classification using convolutional neural networks.

[11] Johnson, J. M., Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. Journal of Big Data, 6(1), 27.

[12] Cui, Z., Chen, W., Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995.

[13] Van Hulse, J., Khoshgoftaar, T. M., Napolitano, A. (2007, June). Experimental perspectives on learning from imbalanced data. In Proceedings of the 24th international conference on Machine learning (pp. 935-942).

[14] Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., Bakshy, E. (2019). Botorch: Programmable bayesian optimization in pytorch. arXiv preprint arXiv:1910.06403.

[15] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.