# A beat classifier for the detection of Premature Atrial and Ventricular Complexes

Bertone Elisa, Peteani Giulia, Scandelli Alice, Veneruso Chiara

*Abstract* – **The aim of the presented work is to create an automatic beat classifier for ECG signals in order to distinguish Normal beats (N), Supraventricular beats (S) and Ventricular beats (V).**

**Different approaches were explored, including Machine Learning (ML), Deep Learning (DL) and hybrid methods. In the end, a hybrid approach was chosen since it was the one that guaranteed to achieve the best performances in terms of precision, recall and f1-score while differentiating pretty well normal beats from abnormal ones.**

## 1. INTRODUCTION

The beating of the heart produces electrical activities which can be measured on the body surface through an electrocardiogram recording (ECG).

The analysis of ECG signals allows to infer very significative information on a patient's health. If an ECG signal shows a heartbeat rhythm which is irregular, either faster (>100 beats/min), or slower (<60 beats/min) than what is physiologically classified as normal, this falls into the definition of cardiac arrhythmia [5]. Arrhythmia can cause several types of consequences: an imminent threat to a patient's life (e.g., ventricular fibrillation and tachycardia), long-term threats, or even death.

Normal cardiac rhythm can be occasionally interrupted by a beat that occurs before the regular time of the next sinus beat, and this is described as a premature beat or premature contraction. The premature beat can be classified into two types depending on the location of the focus, which is different from the Sinus Node (SN):

- Premature Atrial Contraction (PAC) (also known as atrial premature beat (APB)) if its origin is in the atria or the AV node;
- Premature Ventricular Contraction (PVC) (also known as ventricular premature beat (VPB)) if its origin is in the ventricles [1].

The occurrence of a PAC is linked with:

- an abnormal P wave morphology, whose degree of deviation from a normal counterpart depends on the distance between the actual location of the focus and the SN;
- a QRS complex morphology matching that of a normal sinus beat, but for an associated compensatory pause; the interval between the two sinus beats that enclose the PAC is less than the length of two normal RR intervals.

Unlike the PAC, the presence of a PVC:

- almost always prevents the occurrence of the next sinus beat, thus causing a stronger alteration in the rhythm;
- produces a QRS complex abnormally prolonged and with a morphology that deviates considerably from that of a sinus beat [1].

For these reasons, the detection of PVC is an easier task.

## 2. MATERIALS

The data used in this study consist of 2-leads ECG signals from 105 patients, along with the corresponding R peaks positions and annotations (N, S, V labels). The signals were acquired either at 128 Hz or 250 Hz.

An example of annotated signal employed is presented in Fig. 1. From Fig. 1a-1b, it can be noticed that the PAC beat (purple peak) has an R-R interval which is smaller if calculated with respect to the previous beat, and bigger if calculated with respect to the next one. Moreover, in Fig. 1c-1b, it can be seen that a PVC (red peak) is a beat with a completely different morphology and with a different R-R interval too.

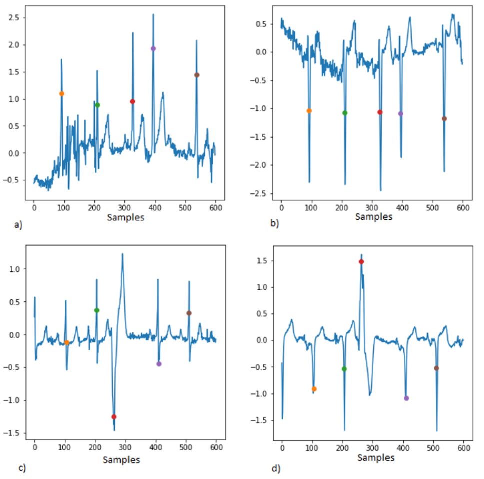

Fig. 1. a) Example of a PAC beat (purple) on lead 1. b) Example of the PAC beat on lead 2. c) Example of PVC beat (red) on lead 1. d) Example of PVC beat on lead 2.

## 3. METHODS

### 3.1 Data Analysis and Preprocessing

Since there were no redundant or missing signals, none of the patients was a priori excluded from the study.

**Class imbalance** As expected, the inspection of the dataset revealed a strong imbalance between the different classes, containing a huge number of normal peaks with respect to the other two classes, as shown in Fig. 2.

In order to reduce a bit this condition, signals coming from patients where all the beats were labelled as 'N' were removed. In this way, the number of patients was reduced from 105 to 91.

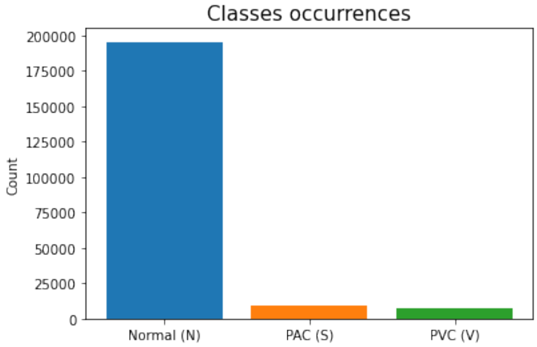

Fig. 2. Classes occurrences

However, even after this removal, a huge imbalance between target classes was still present and different strategies were employed to face this problem, as will be described in paragraphs 4.7 and 5.3.

**Resampling** As already mentioned, the signals included in the dataset were sampled either at 128 Hz or 250 Hz.

However, as will be described in section 4.1, some of the extracted features will be expressed in terms of number of samples. This led to the need of uniforming the sampling frequency: in particular, all signals were resampled to 128 Hz, which was demonstrated to be a sufficient value for ECG sampling frequency when signal analysis is limited to time domain [10].

Of course, R peaks positions were resampled accordingly.

**Filtering** Subsequently, all ECG signals were filtered to reduce the interference of high frequency noise and to remove baseline wander. Different trials were performed varying the order and the cut-off frequency of a Butterworth filter: in the end, by visual inspection (Fig. 3), a filter of order 3 was chosen and frequencies between 2 Hz and 20 Hz were kept [11,12].

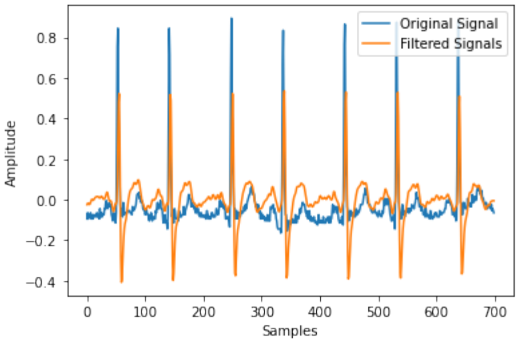

Fig. 3. Example of a signal before and after filtering.

## 4. MACHINE LEARNING

### 4.1 Feature Extraction

In order to make the study the as robust as possible, a total of 69 features describing the characteristics of each heartbeat have been extracted.

In particular, for each heartbeat of the two-leads ECG signals, features relating to both heartbeat intervals and ECG morphology were separately computed. Indeed, due to a significant inter-patient ECG morphology variability, morphological information alone is believed to be not sufficient to classify the ECG heartbeats. Therefore, it is coupled with timing information, which is more invariant among patients, allowing to achieve high classification performance for a huge dataset [3].

Moreover, due to the specific objective of this work, the same features were computed considering windows of different lengths to capture informative variations from neighbor beats.

**Temporal Features** A literature analysis [2,3] of the most significant temporal features led to the extraction of the following information:

- RR intervals: distance of two consecutive R peaks
  - Pre-RR-intervals: the RR-interval between a given heartbeat and the previous heartbeat;
  - Post-RR-intervals: the RR-interval between a given heartbeat and the following heartbeat.
- dRR: difference of consecutive RRs, computed as RR(i+1)-RR(i) considering both pre RR-intervals and post RR-intervals.

Considering two different windows (3 and 15 beats before and after the current beat), the following features were computed:

- Mean RR
- Standard deviation of RRs
- Standard deviation of dRRs
- Percentage of successive interval differences greater than 10, 20, 30, 40, and 50 ms (pNN10, pNN20, pNN30, pNN40, pNN50)
- Root Mean Square of Successive Differences (RMSSD). The RMSSD reflects the beat-to-beat variance in the heart rate and is the primary time-domain measure used to estimate the vagally mediated changes reflected in HRV [4].

In the computation of the RR intervals and dRR, and in their other related features, some NaN values were introduced in the beats located at the beginning and at the end of each ECG signal. For the features computed over windows, NaN values were assigned to features corresponding to the first and last 3 and 15 beats, respectively, when considering 3 and 15 beats window.

**Morphological Features** The morphological features, extracted separately for each of the two leads, are listed hereafter:

- Beats amplitude, computed as the maximum value minus the minimum one for each beat;
- QRS, P wave and PR segment amplitudes;
- QRS, P wave, and PR segment Maximum Cross correlation, using an intra-patient template obtained considering three different windows (4, 20, 80); [2]

- QRS, P wave ad PR segment Lag corresponding to the cross-correlation value considered above. [2]

The identification of ECG segments was not performed using delineation algorithms to identify the onset and offset of each segment. Instead, they were extracted considering different windows centered on the R peaks, using values taken from literature [2]:

– [–300, 40] ms for the P wave segment;
– [–70, 60] ms for the QRS complex;
– [–288, 0] ms for the PR interval;
– [–300, 250] ms for the whole beat.

For each of these segments, 3 intra-patient templates were created (Fig. 4), based on the different windows: 4 beats (2 before and 2 after the current beat), 20 beats (10 before and 10 after the current beat), 80 beats (40 before and 40 after the current beat). Templates were obtained as follows:

1. Align the standardized segments (beat, P wave, QRS complex, PR interval) inside the considered window through cross-correlation.
   Associate to each segment the average of the maximum correlation between the segment itself and the others in the considered window.
2. Discard outliers to obtain a template which is as much as possible similar to the segment of interest in the normal scenario.
   In order to do this, a threshold value of 0.75 was chosen, by trial and error, and all segments characterized by a correlation lower than this value were discarded.
3. Compute the mean among the remaining segments, obtaining the template.

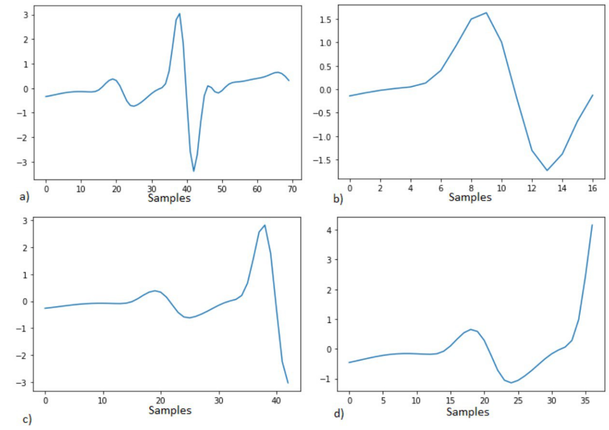

Fig. 4. Examples of: a) Beat template b) QRS template c) P wave template d) PR template.

The next step consisted in computing the cross correlation between the template and each corresponding segment (beat, QRS, P wave, PR segment). In this way, two features for each considered window were extracted, namely the maximum cross correlation and the corresponding lag.

Finally, a DataFrame containing 212673 beats, each with 69 features associated, was created.

### 4.2 Exploratory Data Analysis

An analysis of features distribution was carried out to evaluate the necessity of scaling, outlier removal and applying transformation techniques.

As expected, the histograms visualization revealed that features had different orders of magnitude. An example with only 3 of the extracted features is shown in Fig. 5.

Thus, scaling operation was carried out immediately after the train-test split, as will be described in section 4.6.

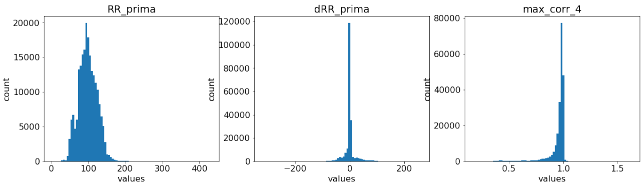

Fig. 5. Histogram representation of three of the extracted features.

To perform a deeper exploration of the extracted features, their distributions were separately plotted for the different classes with the aim of investigating if each single variable was, alone, able to significatively differentiate samples belonging to different classes. Some of them can be observed in Fig. 6.

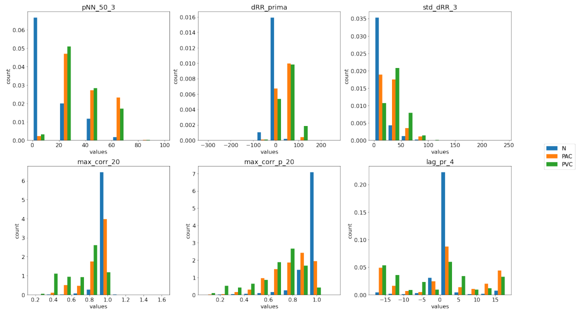

Fig. 6. Distributions of some features for the different classes.

Looking at these graphs, it is intuitive that PVC class can be distinguished pretty easily from Normal one. On the contrary, the distribution of PAC class basically overalps with the Normal one for many features, making the classification very challenging.

For example, it can be verified that evaluating the maximum correlation on the whole beat ("max_corr_20" in Fig. 6) between each beat and an intra-patient template (built to be representative of a normal beat, as described in section 4.1, so that the distribution is skewed towards 1 for class N), PVC ones almost never assume value equal to the ones relative to Normal beats, while PAC beats do since their QRS complex does not present a significant morphological distortion.

Instead, evaluating the maximum correlation on the P-wave segment ("max_corr_p_20") it can be inferred that, making a comparison with the previously commented ("max_corr_20"), PAC beats show a more spread distribution, which can be interpreted knowing that the slight morphological distortion in PAC beats regards only the amplitude of the P-wave, which is more (or less) significantly reduced with respect to the normal case as the focus is farther (more near) from SN.

Considering, instead, some temporal features, as pNN50 and dRR in Fig. 6 (“pNN_50_3” and “dRR_prima”), both PVC and PAC show a significantly different distribution with respect to beats belonging to the normal class, reflecting the alteration of the rhythm that is present for both these pathological beats.

### 4.3 Handling abnormalities

By inspecting the originally extracted features through boxplots, it was noticed that some RR intervals assumed too high values (Fig. 7).

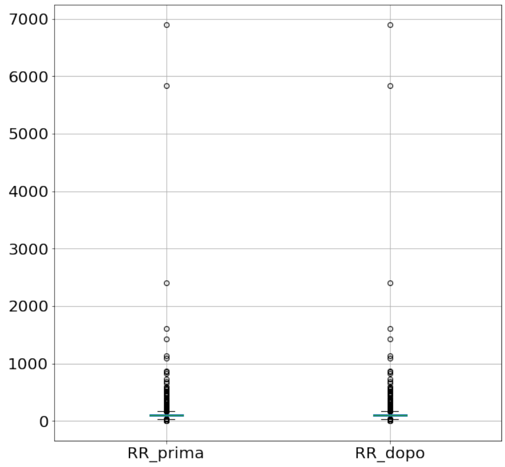

Fig. 7. Boxplot representation of temporal features RR_prima and RR_dopo.

A focused signals’ inspection led to the result that these outliers were generated after feature extraction procedure because some peaks were not annotated, as it is shown in Fig. 8.

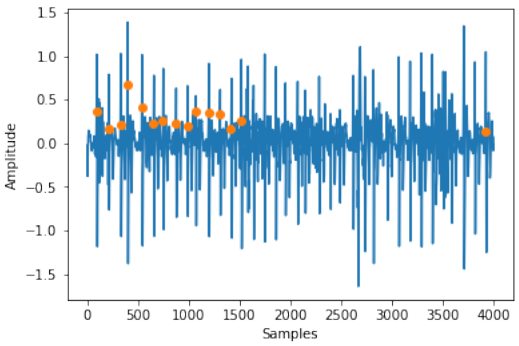

Fig. 8. Example of ECG signal wrongly annotated

The choice was to not intervene on the peak detection algorithm, which was out of the scope of the presented work, and to focus instead the effort on how to handle these abnormalities generated after the feature extraction phase. Being sure that these were resulting from actual errors and that were not reflecting a particular condition that could be useful to take into consideration, it was decided to treat them as missing values (NaN).

Different possibilities were considered, such as the one of not including the beats affected by these exceptions in the final dataset. This could have been a valid choice since the total number of beats, over all the signals, classified as outliers for the described reason was of 56, on a total of 212673, meaning only the 0.024%. As a consequence, their elimination would have caused a not significant reduction of sample size, thus not bringing to a decrease of the statistical power of the classifier, as it was formally proved by a further trial which will be described in section 4.10.

However, a solution applicable also to the test set, in which there was the need to associate a label to each beat without the possibility of excluding some of them, was needed. In this perspective, a sophisticated possibility initially taken into consideration was to perform multiple imputations with a repeated stochastic regression method, subsequently generating multiple datasets and then combining the different parameters estimates to reach a final highly unbiased estimate. However, due to the very reduced percentage of outliers, this method was not explored after an evaluation of the balance between the computational cost required by this complex procedure and the potential advantages.

In the end, a simpler single imputation method was chosen, taking advantage of the knowledge of the specific field. Therefore, a normal and a maximum value for the variable of interest were taken from literature: whenever the RR interval computed was above a fixed maximum value (defined as about 3428,6 ms, value referred to the occurrence of a refractory pause after a PAC in an extreme case of bradycardia [2]), it was substituted with a normal value, assuming a heartrate of 60 beats per minute.

To deal with the NaN introduced for all the temporal features computed over windows of length 3 and 15, as well as for RR and dRR of initial and final beats of each ECG signal, the same reasoning described for handling abnormalities in RR features due to undetected peaks was done.

The final decision consisted in a single imputation method using the intra-patient mean of that specific feature. In this case, the risk of introducing bias was slightly higher, since the number of beats containing NaN was around 0.8%: therefore, an analysis to compare the performances of the trained model with all the beats (after performing mean imputation) and without these beats was done, and it will be described in section 4.10.

### 4.4 Feature Selection

A dimensionality reduction through feature selection was performed in order to identify and remove irrelevant and redundant information before the learning phase, and thus to allow an overall higher efficiency.

In this regard, several methods could have been employed.

Concerning filter methods, a correlation-based feature selection was performed using the following criterion: if the correlation between 2 features is low (correlation coefficient less than 0.05), the characteristics are independent, whereas if it is high (correlation coefficient greater than 0.95) the two variables are not independent and, therefore, it is possible to predict one from the other.

Consequently, when 2 features were highly correlated, among the two, the one which correlated more with all other features was discarded, keeping instead the one with a higher degree of novelty in terms of information content.

In addition, another feature reduction step was performed based on variance analysis: the variance of each feature was computed and features exhibiting insignificant variability (variance lower than 0.03) were removed.

After feature selection phase, only 31 of the extracted 69 features were kept.

### 4.5 K-fold cross validation

To choose the best machine learning model class for fulfilling the assigned task, k-fold cross validation method was employed in order to reduce the luck component in the train-test split, which otherwise could severely influence models' performance assessment.

Due to the specific clinical problem faced, a more stringent version than the classical k-fold cross validation was implemented: in particular, the function GroupKFold, already included in sklearn module, was employed to ensure that all the beats of a specific patient necessarily fell in only one of the folds. By doing so, it was possible to guarantee that none of the subsequently built k models could be both trained and then tested on the beats of a same patient, which otherwise would bring to a dangerous bias and overestimation of model's performances.

K was chosen equal to 13, searching for a tradeoff between a too high k's value, leading to big computational effort to train k different models, and a too low one, bringing to an insufficient cardinality of the training set resulting from the single split (in fact k-1 folds are merged to constitute the training set and the remaining one is used as test set).

On the basis of the explained 13-folds subdivision, for each investigated machine learning model class, 13 slightly different models were obtained (according to which was the combination of the 12 folds used for their training) and employed for predicting the labels of their corresponding test set. These output labels were then concatenated into a single vector of predictions which, after a proper sorting, was compared to the one of the true labels for the computation of the metrics used for model evaluation, as will be described in section 4.9.

### 4.6 Feature Scaling and Transformation

The next operation performed on the extracted features in each train set consisted in feature engineering, carried out in order to bring them on the same scale, since it is known that it significatively affects the performances of most of ML models, that in fact could give different relevance to features depending on their order of magnitude.

Training data were rescaled using a standardization technique, hence bringing them to follow a normal distribution with 0 mean and unitary standard deviation.

The very same statistical parameters computed on the train set were then used to rescale the corresponding test set data: it is in fact a good practice to fit the scale on training data and then use it to transform the test set in order to avoid data leakage during testing phase.

Looking at the histograms, it was observed that some features, such as the maximum correlation on both lead1 and lead2, for some of the segments under analysis, had an exponential behavior. Thus, a logarithmic transformation was applied to these features in order to obtain a normal distribution. However, this did not lead to any improvement and, therefore, the initial features were kept.

### 4.7 Facing imbalance in training set

Given the strong classes imbalance, different possibilities were investigated in order not to overfit the model, that could be biased toward predicting the most frequent class. Both over-sampling and under-sampling procedures were explored, but the methodology that allowed to reach the best model performances was using class weights.

These were used as weighting factors in the loss function which must be minimized during the training phase: thus, they were computed as the inverse of the frequency of a class label in the training set in such a way to penalize more the errors that the model performs in predicting the minority class.

### 4.8 Train-validation split

The dataset, already deprived from the fold which will act as test set for a specific split (as described in section 4.5), was further subdivided in train and validation sets. Then, the validation set is employed to perform an hyperparameter selection through a grid search.

During this work, a peculiar form of k-folds cross-validation [6], with k=10, was ad hoc designed to guarantee both that the beats of a patient fell in one fold only, as already described in section 4.5, but also that there was stratification in the split to preserve the ratio between label classes in each created fold. This choice, even if more computationally costly than the splitting procedure adopted for train-test sets, was here retained necessary to ensure a robust choice of the best final combination of hyperparameters.

### 4.9 Model Selection

Several Machine Learning models were tested:

- Decision Tree
- K-nearest neighbors (KNN)
- Support Vector Machine (SVM)
- Random Forest
- Extra Tree
- Adaboost with Decision Trees
- Multi-layer Perceptron (MLP)

The training was performed minimizing the macro f1-score, which appeared to be the best metric when dealing with an imbalanced dataset as it treats all classes equally.

All the models were evaluated based on model precision, recall and f1-score, computed separately on the 3 classes.

The performance metrics were computed comparing the true known labels with the predictions' vector obtained as the concatenation of the outputs of the 13 models resulting from the 13 different train-test splits according to k-fold cross validation method. In turn, each of these 13 models was the result of a grid search procedure performed again with k-fold cross validation for train-validation split, with k=10, ensuring stratification.

Rather than designing an ensemble of these 13 models, for seek of simplicity, it was preferred to define the final model of each model class by choosing the most frequent set of hyperparameters among the 13 resulting ones and then retraining this latter on all the data at disposal.

The results of these classifiers, along with the selected hyperparameters, are shown in Table I.

### 4.10 Results

The choice of the best ML classifier was not only based on the performances in terms of precision, recall and f1-score, which were comparable among many of the tested models. Indeed, another aspect which was taken into account was the specific clinical problem: the reasoning underlying the choice was to differentiate as much as possible between healthy and pathological beats.

Among the tested model, the Extra Tree classifier was the one which guaranteed the best tradeoff between all these aspects, as can be seen in Fig. 9.

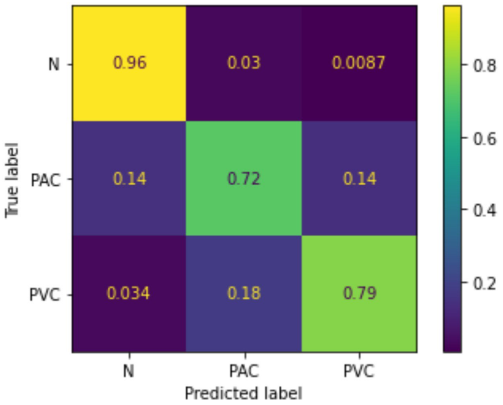

Fig. 9. Confusion Matrix of Extra Tree Classifier.

After the selection of the best ML classifier, some additional analyses were performed taking it as reference model, fixing all the hyperparameters, to confirm some reasonings done in data preparation phase.

The same model's performances were reached in fact when eliminating from the dataset the beats immediately before or after the not annotated ones, that was expected since the reduction in sample size was so negligible that it didn't affect the statistical power of the classifier.

The same conclusion was reached also when training the reference model eliminating from the dataset the rows of the 15 initial and final beats, for which the intra-patient mean substitution had been performed.

## 5. DEEP LEARNING

### 5.1 Beat Windowing

When dealing with a deep learning approach, the feature extraction block is data driven, meaning it is performed by the neural network, which only receives the single beats as input.

The beats have been extracted from the pre-processed signals (section 3.1) by taking a window which spans from –2s to +2s starting from the R peak of each beat. This window was chosen empirically. Indeed, by trials, it has been noticed that with a smaller window including only the beat to be classified, the performance of the models was lower.

This led to the decision of considering a bigger window, which also includes few previous and following beats (Fig. 10) allowing definitely better performances: this can be explained by the fact that a larger window contains useful information about anomalies of pathological heart beats, such as the distance between 2 consecutive R peaks.

To deal with the problem of the first and last beats of each signal, where the number of samples was not sufficient to extract the full window, a padding operation has been performed: finally, a window of 512 samples for each beat has been extracted and fed individually to the network.

The 2 leads were fed separately to the network as well.

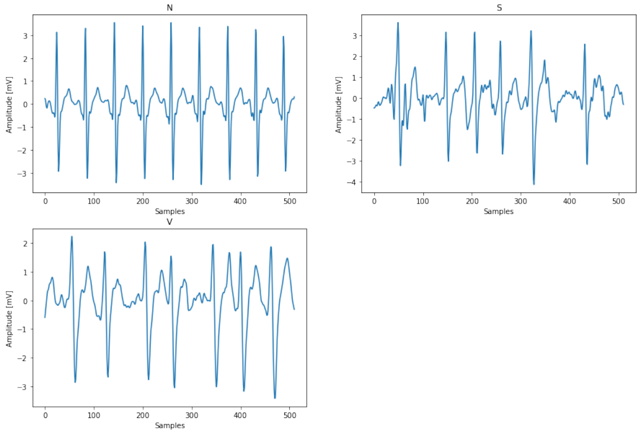

Fig. 10. Example of input window given to the deep learning model in the three conditions: Normal (N), PAC (S), PVC (V).

### 5.2 K-fold cross validation

The same procedure described in section 4.5 has been applied.

Differently from what has been described in section 4.8, when dealing with deep learning models, the choice of the hyperparameters was not based on a rigorous grid search in order to avoid a too elevated computational effort.

In this case, the validation set used for monitoring models' performances during training, as well as for overfitting prevention through early stopping technique, has been extracted with a simple train-validation split with proportion 0.9 - 0.1 to ensure enough data for training, always paying attention to have beats belonging to the same patient fall in only one of the two sets.

### 5.3 Class imbalance

As shown in section 3.1, the extracted beats are very unbalanced with respect to the three labels of interest for classification purpose. If no countermeasures were taken, the models trained with this kind of data would learn much more from the majority class, favoriting it during the predictions, which is the worst-case scenario for diagnosis purposes.

To cope with this issue, an undersampling of the most frequent classes has been performed to balance class

TABLE I
Machine learning models with selected hyperparameters and performance metrics

| Model | Hyperparameters | Precision | | | Recall | | | F1-score | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | N | S | V | N | S | V | N | S | V |
| Decision Tree | ‘criterion’: ‘entropy’<br>‘max_depth’: 7<br>‘min_samples_leaf’: 5<br>‘min_samples_split’: 5 | 0.97 | 0.47 | 0.60 | 0.98 | 0.43 | 0.59 | 0.98 | 0.45 | 0.60 |
| KNN | ’n_neighbors’: 55 | 0.97 | 0.55 | 0.74 | 0.99 | 0.43 | 0.62 | 0.98 | 0.49 | 0.67 |
| SVM | ‘C’: 100<br>‘gamma’: 0.0001<br>‘kernel’: ‘rbf’ | 0.99 | 0.47 | 0.60 | 0.95 | 0.77 | 0.79 | 0.97 | 0.58 | 0.68 |
| Random Forest | ‘criterion’: ‘entropy’<br>‘max_depth’: 30<br>‘min_samples_leaf’: 5<br>‘min_samples_split’: 10<br>‘n_estimators’: 150 | 0.99 | 0.59 | 0.70 | 0.98 | 0.66 | 0.79 | 0.98 | 0.63 | 0.74 |
| Extra Tree | ‘criterion’: ‘entropy’<br>‘max_depth’: 20<br>‘min_samples_leaf’: 7<br>‘min_samples_split’: 15<br>‘n_estimators’: 100 | 0.99 | 0.49 | 0.67 | 0.96 | 0.71 | 0.79 | 0.98 | 0.58 | 0.72 |
| Adaboost | ‘max_depth’: 3<br>‘min_samples_leaf’: 5<br>‘learning_rate’: 0.1<br>‘n_estimators’: 120 | 0.98 | 0.63 | 0.77 | 0.99 | 0.60 | 0.70 | 0.98 | 0.62 | 0.73 |
| MLP | ‘activation’: ‘relu’<br>‘alpha’: 0.0001<br>‘hidden_layer_sizes’: (10, 30, 10)<br>‘learning_rate’: ‘adaptive’<br>‘solver’: ‘adam’ | 0.98 | 0.55 | 0.72 | 0.98 | 0.52 | 0.67 | 0.98 | 0.54 | 0.70 |

distribution, as it is shown in Fig. 11: randomly, a number of samples equal to those belonging to minority class (‘PVC’) has been extracted from the majority classes (‘Normal’ and ‘PAC’) for every k-th train set built according to the k-fold procedure.

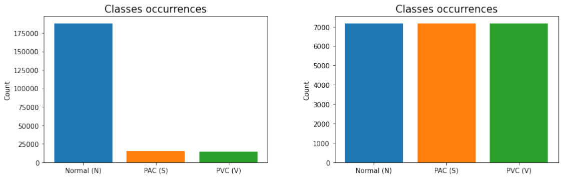

Fig. 11. On the left the original distribution of labels in the training dataset. On the right the balanced distribution.

### 5.4 Model architecture

The proposed model is a Convolutional LSTM with an attention layer.

Combining the knowledge acquired in the biomedical field on the specific signals of interest with the one on deep learning models presented in literature, it has been considered the necessity of employing:

- convolutional layers, widely used for morphological analysis due to their unique ability of capturing position and translation invariant patterns [7] and to their proved capability to extract useful information even from noisy signals [8].
- LSTM units to capture the temporal dynamics through their ability to selectively remember or forget information. The bidirectional LSTM implementation is of particular interest, giving the possibility to integrate the hidden state vectors of LSTM units processing ECG fragments both in forward and in reverse direction [9].

The architecture, shown in Fig. 12, consists of a convolutional layer with ‘ReLu’ nonlinear activation function, kernel size 1 × 1, followed by a Dropout layer (0.3 of probability), which allows to prevent overfitting.

Then, a Bidirectional LSTM layer with 256 units is added, followed by another Dropout layer (0.3 of probability).

At this point, the attention is implemented: attention is a mechanism combined in the Neural Networks with recurrencies, allowing them to focus on certain parts of the input sequence when predicting the output, enabling easier and better quality learning. Attention has been implemented by means of a function which takes as input the output of the previous layer, applies a dense layer which computes the SoftMax and finally multiplies each input value for the

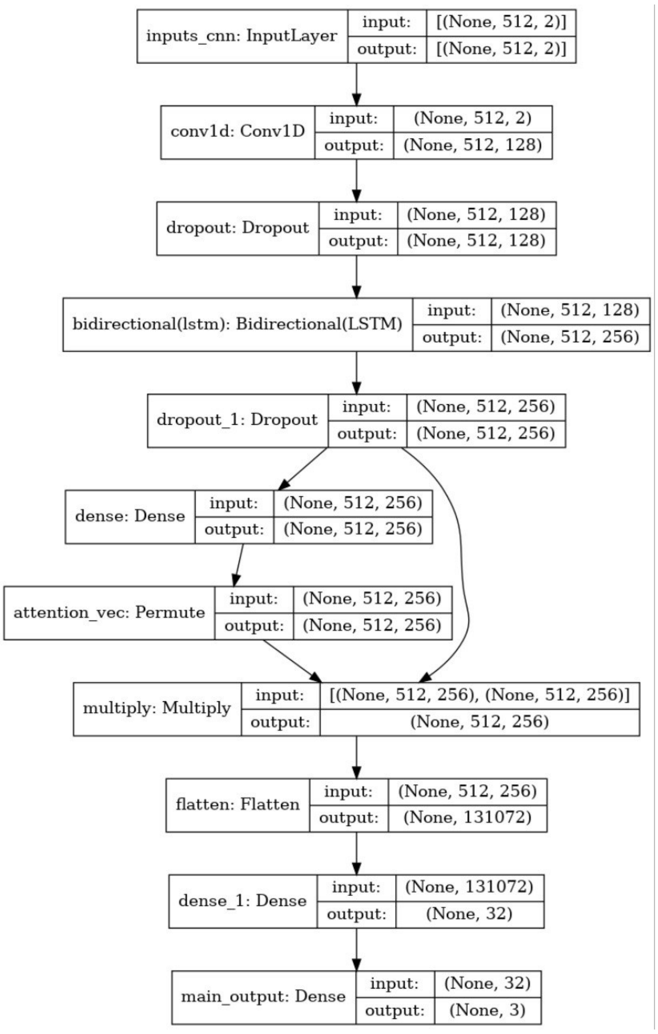

Fig. 12. Deep Learning Model architecture.

corresponding score of the SoftMax, weighting them.
The objective here is to allow the network to learn the best way to distribute attention: in fact during the training phase the network's parameters are optimized with the aim of guaranteeing a proper computation of the percentage attention weights resulting in output from the SoftMax layer.

Finally, a Flatten layer followed by 2 Dense layers (the last with a number of outputs equal to the number of classes, 3) are added.

The three output neurons make the final classification using a SoftMax activation function.

To prevent overfitting occurrence, together with the insertion of dropout layers, an early stopping technique was employed, which consists in using the validation set to monitor the model performance during the training phase, to eventually stop it if the validation loss stops decreasing.

The model has been trained k = 13 times over 25 epochs, using each time the correspondent k-1 folds. The training was performed by minimizing the Categorical Cross-entropy loss with ADAM optimizer, which is able to adjust the learning rate (initially set to 0.0001) based on the history of the gradient to both speed up the learning and increase the performance.

Hyperparameters, such as the number of epochs, number and dimension of filters, dropout probabilities and initial learning rate value were chosen after several manual trials.

The predictions made on each k-th test set were, at the end, concatenated. The resulting confusion matrix is shown in Fig. 13.

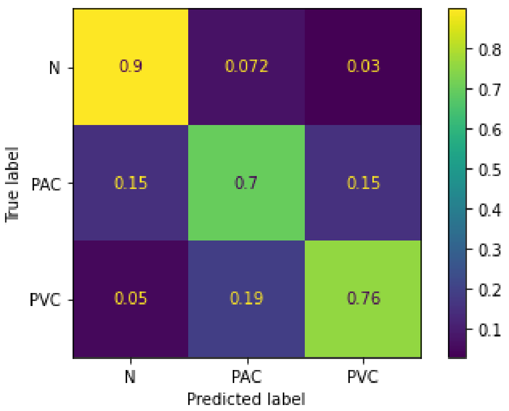

Fig. 13. Confusion Matrix of the Deep Learning model.

Results, shown in Table II, have been evaluated in terms of the model precision, recall and f1-score on the 3 classes separately.

**TABLE II**
**Deep Learning Model results**

| | Precision | Recall | f1-score |
|---|---|---|---|
| N | 0.99 | 0.98 | 0.94 |
| PAC | 0.30 | 0.70 | 0.42 |
| PVC | 0.45 | 0.76 | 0.56 |

## 6. HYBRID APPROACH

Eventually a hybrid approach was developed, adding to the deep learning network described in the previous paragraph a Dense layer whose inputs are the temporal features manually extracted during the development of the ML models.

The choice of including in this model only the temporal features was driven by the desire to try to improve the capability of the previously developed DL model in discriminating beats belonging to normal class from PAC ones, thus the addition of the morphological features was retained not strictly necessary for this specific purpose.

The procedure was analogue to the DL approach: the dataset, containing the beats on windows of 4 seconds and 17 temporal features (selected in the same way described in section 4.4), is split into 13 couples of train and test folds that are used to train 13 models in order to assess the performances of the network architecture under examination in a robust way.

During the training, the features contained in the current train and test folds are scaled using StandardScaler.

The training data is then undersampled to have a balanced labels' distribution, and divided into train and validation sets.

The used network is shown in Fig. 14. It contains two branches that concatenate before a Dense layer and the softmax layer: one branch takes as input the two-leads beat signal, so it has input shape (512, 2), and its architecture is identical to the one of the network described in the deep learning paragraph; the other branch takes as input the 17 temporal features and is simply composed of a Dense layer with relu activation.

After each training, the predictions over the k-th test set were computed: the final confusion matrix is shown in Fig. 15 and the results are displayed in Table III.

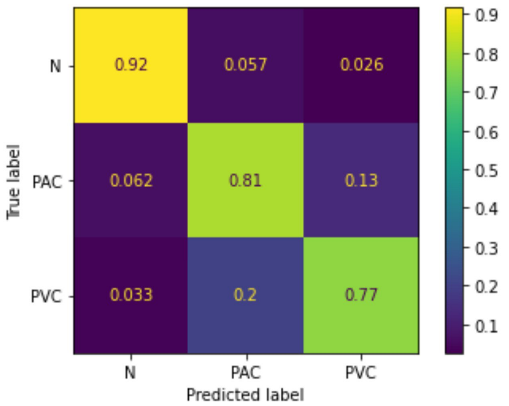

Fig. 15. Confusion Matrix of the Hybrid model.

**TABLE III**
**Deep Learning Model results**

| | Precision | Recall | f1-score |
|---|---|---|---|
| N | 1.00 | 0.92 | 0.95 |
| PAC | 0.38 | 0.81 | 0.51 |
| PVC | 0.49 | 0.77 | 0.60 |

## 7. DISCUSSION

With the aim of creating an automatic beat classifier for ECG signals in order to discriminate between normal, supraventricular and ventricular beats, the initially explored possibility was the one of extracting hand crafted features, thus following a machine learning model development approach.

This had the advantage of allowing a deep exploitation of a priori knowledge on ECG signal, in particular on the peculiarities of PAC and PVC beats, whose features' distribution could be accordingly interpreted in a reliable way, as done in paragraph 4.2.

However, models which are known to be characterized by a high interpretability and which could truly be advantageous in the clinical field (e.g. to provide useful reliable information to a clinician which, on that base, could perform informed decisions), like the decision tree classifier, were actually found to be not able to discriminate between the classes of interest.

The best-found performing ML model was instead an extra tree classifier that, being an ensemble model, does not guarantee an easy interpretation of produced outputs. Moreover, its capability to distinguish normal beats from abnormal ones, which is the most relevant need from a clinical point of view, is not optimal since a significant amount of PAC beats were predicted as Normal (as testified by a PAC's recall of 0.71).

To overcome this limitation, going however toward the direction of working with a black box, also a deep

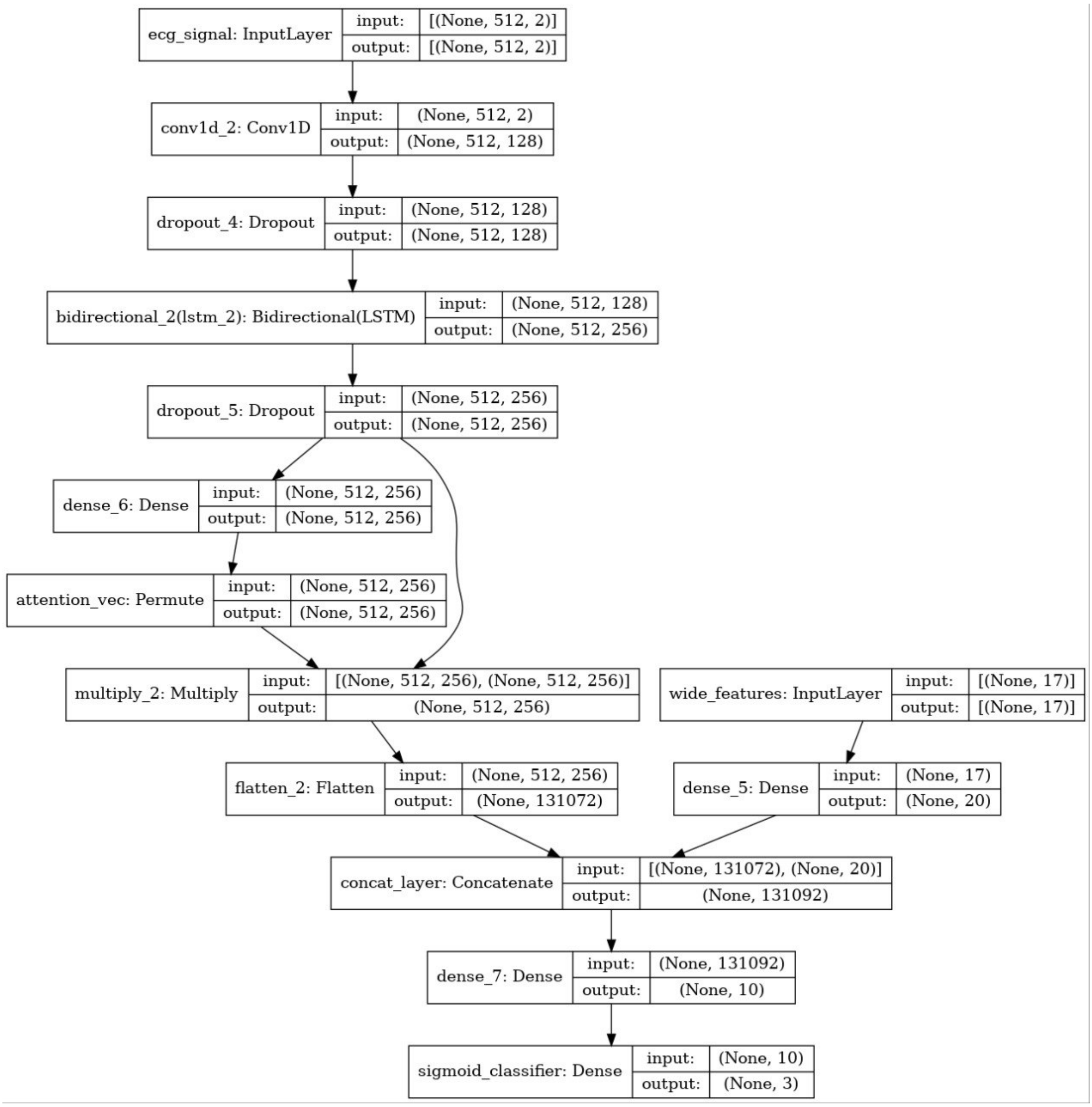

Fig. 14. Hybrid Model architecture.

learning model development approach was investigated, mixing different neuron layer types to automatically extract information on both morphological and time beats characteristics.

The final model, resulting from a manually executed hyperparameters search, revealed however the same limitation of the best developed ML model in terms of classifying a significant amount of PAC beats as normal ones (with an identical value of PAC's recall).

The improvement with respect to the ML model can however be captured in another perspective: the DL model, once ended the computationally costly training phase, has the great advantage of not requiring the computation of hand-crafted features for the new beats to be classified. This means it could potentially be employed in on-line applications (e.g. in intensive care units or in wearables), admitting the usage of a real time peak detection algorithm and beats windows extraction.

Since the upon described models were not completely fulfilling the main clinical purpose, one last approach was investigated: a hybrid model, combining the great capability of DL models to extract useful information from raw data with the high reliability of manually extracted features. This model resulted much safer from a clinical point of view since it showed an optimal capability in distinguishing normal beats from abnormal ones, assuring a PAC's recall of 0.81 and only making some misclassifications between the pathological classes.

## 8. CONCLUSIONS

In conclusion, between the developed models, the one chosen for performing predictions on the final test set was the hybrid one, due to the lack of necessity of performing a real time labeling and preferring instead a higher reliability level of the features on which the classification is based, as well as the accomplishment of the most relevant clinical purpose. [1]

The main limitation of the final proposed model is related to a not completely satisfactory capability to discriminate, among abnormal beats, between PAC and PVC ones. In a future work the efforts can be thus focused on searching for a better combination of hand crafted features to be given as input to the respective branch of the hybrid model. For sure, this search should include also the morphological features, that are known to be strongly discriminative for the classes of interest. In this perspective, sophisticated techniques for feature selection could be experimented, as wrapper methods with forward, backward or hybrid stepwise selection.

## REFERENCES

[1] Sraitih M, Jabrane Y, Hajjam El Hassani A. An Automated System for ECG Arrhythmia Detection Using Machine Learning Techniques. J Clin Med. 2021 Nov 22; 10(22):5450

[2] García-Isla G, Mainardi L, Corino VDA. A Detector for Premature Atrial and Ventricular Complexes. Front Physiol. 2021 Jun 16;12:678558.

[3] Das MK, Ari S. Patient-specific ECG beat classification technique. Healthc Technol Lett. 2014 Sep 26;1(3):98-103.

[4] Shaffer F, Ginsberg JP. An Overview of Heart Rate Variability Metrics and Norms. Front Public Health. 2017;5:258. Published 2017 Sep 28.

[5] Fu DG. Cardiac Arrhythmias: Diagnosis, Symptoms, and Treatments. Cell Biochem Biophys. 2015 Nov;73(2):291-296.

[6] https://github.com/scikit-learn/scikit-learn/issues/13621

[7] Oh SL, Ng EYK, Tan RS, Acharya UR. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. Comput Biol Med. 2018 Nov 1;102:278-287.

[8] Yanmin Qian, Mengxiao Bi, Tian Tan, Kai Yu, Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu. 2016. Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition. IEEE/ACM Trans. Audio, Speech and Lang. Proc. 24, 12 (December 2016), 2263–2276.

[9] Cheng, J., Zou, Q. Zhao, Y. ECG signal classification based on deep CNN and BiLSTM. BMC Med Inform Decis Mak 21, 365 (2021).

[10] Kwon O, Jeong J, Kim HB, Kwon IH, Park SY, Kim JE, Choi Y. Electrocardiogram Sampling Frequency Range Acceptable for Heart Rate Variability Analysis. Healthc Inform Res. 2018 Jul;24(3):198-206. doi: 10.4258/hir.2018.24.3.198. Epub 2018 Jul 31.

[11] Kaya, Yasin. (2018). Classification of PVC Beat in ECG Using Basic Temporal Features. Balkan Journal of Electrical and Computer Engineering. 10-14.

[12] Alfaras, Miquel Soriano, Miguel Ortín, Silvia. (2019). A Fast Machine Learning Model for ECG-Based Heartbeat Classification and Arrhythmia Detection. Frontiers in Physics. 7.