PPG Signals Classification

Francesco Caserta, Dario Cavalli
Applied AI in Biomedicine, 2024
Politecnico di Milano, Italy

1 INTRODUCTION

Classifying abnormal heartbeats poses a significant challenge in the analysis of PPG signals. These signals are highly susceptible to errors from various sources such as movement, introducing complexity in their interpretation. Nevertheless, given the ease and cost-effectiveness of acquiring PPG signals, it becomes crucial to develop accurate methods for classifying heartbeats to detect potential cardiac pathologies.

The aim of this paper is to showcase the primary techniques used for the accurate classification of heartbeats in a PPG signal. This encompasses the criteria employed for extracting individual heartbeats, various preprocessing techniques to handle noise, and the creation of deep learning models for accurate classification.

Furthermore, in line with common practices in the biomedical field, especially in the context of an initial screening for identifying potential cardiac pathologies, it is of paramount importance to pay attention during the design of such classifiers. This includes aspects like class balancing and setting thresholds for the final classification. Special emphasis is placed on minimizing false positives (Heartbeats classified as healthy when they are actually pathological) by adapting metrics and data to maximize recall, thereby enhancing the system's sensitivity in detecting significant events.

2 DATASET PREPARATION & VISUALIZATION

2.1 Dataset Composition

The dataset provided consists of PPG signals collected from 105 different patients, each with an average length of approximately 30 minutes. Among these, 62 were recorded at a frequency of 128Hz, while the remaining 43 were recorded at a frequency of 250Hz. The length of PPG signals recorded at a frequency of 250Hz is consistently 450,000 samples for all patients, while for the others, it ranges on average between 230,200 and 230,400 samples.

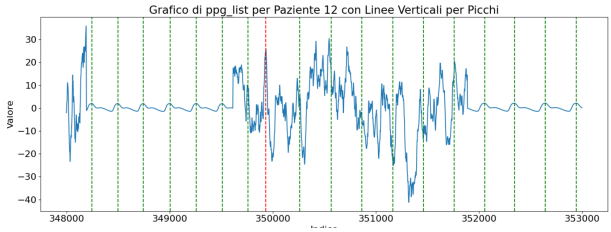

Figure 1: PPG signal with peaks

Associated with each patient's PPG signal is a file containing the positions of the peaks in the signal, and another file containing a label associated with each peak (N for regular beats, S for supraventricular, and V for ventricular beats).

Upon initial exploration of the data, it was observed that the array of peaks for two patients was not ordered. Given the uncertainty about whether the corresponding labels associated with the peaks were ordered or not, it was decided to discard the signals from these 2 patients. Consequently, a usable dataset with information from 103 patients was obtained.

2.2 Downsampling & Upsampling

To achieve a consistent dataset across different patients, it was necessary to first standardize all signals to the same frequency, taking care to adjust the position of the corresponding peak.

We experimented with both approaches, and the performance obtained was similar for both. In the end, we chose upsampling to 250Hz.

Upsampling to bring the signals from 128Hz to 250Hz was performed by applying simple linear interpolation point by point. Subsequently, we updated the positions of the peaks and manually verified their correspondence by analyzing the plot of the signals.

2.3 Dataset Visualization

2.3.1 Noise in Signals. After plotting some signals, we noticed significant noise in various segments of the signal. We delved deeper into the analysis by plotting

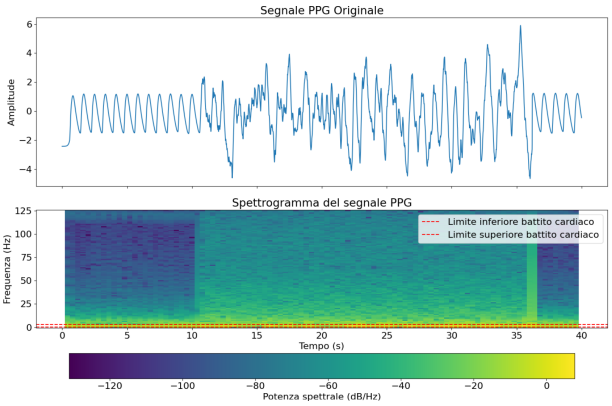

Figure 2: Spectrogram

the spectrogram associated with the signal (see image), using the Short-Time Fourier Transform (STFT) with the following parameters:

-Sampling frequency (fs): 250Hz

-Number of samples per segment (nperseg): 200

-Overlap between segments (noverlap): 100

From the spectrogram (see Figure 2), it becomes immediately apparent which portions of the signal are affected by noise. These segments exhibit a much higher spectral power above 5Hz (the graph appears clearer, tending toward yellow).

Subsequently, we will use this characteristic of the spectrogram to distinguish between heartbeats that contain only noise and those that do not (see Chapter 3.2).

Recognizing the evident influence of noise, we experimented with various approaches to minimize its impact on the signals. Although noise began to significantly affect the signals beyond 5 Hz, we explored the use of a Butterworth filter with different configurations. However, it's important to note that this approach was considered but ultimately not employed in the final analysis.

2.3.2 Class Imbalance. Subsequently, we grouped all the labels from all peaks of all patients and observed their distribution: 'N': 226,448, 'S': 9,691, 'V': 7,994. In percentage, class 'N' corresponds to 92.8% of the total classes, 'S' to 4%, and 'V' to 3.3%. (see Figure 3)

It can be concluded that the distribution between healthy and unhealthy heartbeats is heavily imbalanced (favoring healthy beats), while classes 'S' and 'V' are more balanced with each other. Additionally, we observed how this distribution varied for each individual

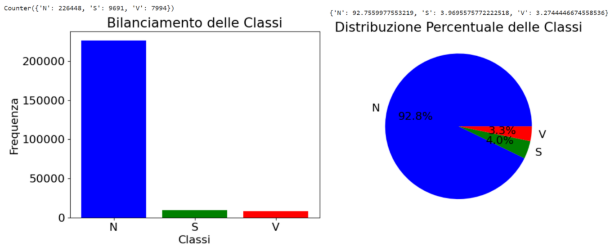

Figure 3: Overall Labels distribution

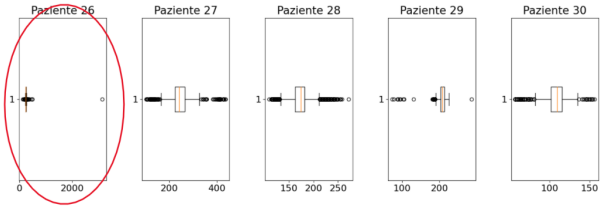

Figure 4: Peak Distance Boxplot per patients

patient, noting that for the majority of patients, healthy beats account for over 96% (including patients with exclusively healthy beats). However, for a smaller subset of patients (around ten), the percentage of unhealthy beats reaches and exceeds 20%.

2.3.3 Peaks distance. We visualized a boxplot for each patient and one encompassing all patients, representing the distances between two consecutive peaks. It is observed that the majority of patients have peak-to-peak distances around 200 ± 50. However, it is evident that some patients exhibit outliers (see Figure 4), characterized by significantly larger distances (see patient 64 in the Colab), likely due to annotation issues. In subsequent implementations, we took these outliers into account to prevent data distortion.

Furthermore, we visualized boxplots on the total distance between the previous and subsequent peaks for all patients, divided by label (N, S, V). It emerges that the average distances are N: 351.11, S: 288.41, V: 318.46, suggesting that the distance between the previous and subsequent peak of a heartbeat could be a good indicator for distinguishing between classes.

2.4 TRAIN, VALIDATION, TEST

Before extracting beats from the signals of the 103 patients, we divided them into training (72 patients), validation (15 patients), and test (16 patients) sets. We opted for patient-wise splitting to prevent beats from

the same patient being distributed across train, validation, and test sets, which could introduce bias into performance metrics.

In splitting the patients, we endeavored, as much as possible, to maintain the proportions of the label distribution (N, S, V).

3 BEAT EXTRACTION & PREPROCESSING

3.1 Beat Extraction

One of the biggest challenges we faced was the extraction of beats from the signal. Since the classification models we used required a fixed-length input, we had to choose the most suitable length for the window corresponding to the heartbeat and how to position it relative to the peak of the beat.

We followed two approaches:

3.1.1 Fixed Window 200: Initially, based on our observations of the signals and peak distances mentioned earlier, we noticed that the average distance between two peaks was 200, indicating that, on average, a beat lasts for 200 instances. Additionally, observing the position of the peak relative to the beat, we found that the peak is typically at 1/3 of the signal length. Based on these observations, we decided to use a window of length 200, positioned approximately at 1/3 of the peak (taking the 66 instances before and 134 instances after the peak). However, this approach did not yield satisfactory results immediately, leading us to abandon it early on.

Possible reasons for this approach not performing well include:

- The value 200 is associated with an average value, meaning that this window often captured only a portion of the beat or included both the preceding and succeeding beats.
- Since the peak-to-peak distance is one of the main discriminants between healthy and abnormal beats [6], this approach often missed the preceding and succeeding peaks, depriving the network of these important features.

3.1.2 Peak-to-Peak portion: The final choice was to extract the entire portion between the preceding and succeeding peaks, up to a maximum of 400 instances (window set to 400). If the portion was longer, it was trimmed equally from both ends; if it was shorter, it was padded equally from both ends. This solution provided better results (see Figure 5), which can be reviewed in the following sections.

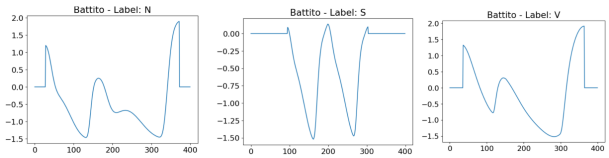

Figure 5: Extracted Beats per each class

3.2 Dealing with Noise

As observed in section 3.3.1, by examining the signals and analyzing the spectrogram, it becomes evident that they contain a significant amount of noise, consistent with what is reported in the literature, given the sensitivity of the acquisition methods.

Once the beats were extracted, we were able to classify whether the signal was noise-free or subject to noise. To achieve this, starting from the data extracted from the spectrogram, we set a threshold on the sum of the powers of frequencies greater than 5Hz (excluding those of the heartbeat). The threshold was manually set at 0.1 dB/Hz, based on observations from the data. Therefore, all beats surpassing this threshold were classified as noise and discarded. Note: this threshold was intentionally set very high (manually, we observed that a noise-free beat rarely exceeds 0.5 dB/Hz) to avoid discarding noise-free beats belonging to the S and V classes (of which we already had few data).

3.3 Fighting Class Imbalance

The training set obtained in this way, as commented earlier, still exhibits a significant imbalance between class N and the others. Initially, we attempted to address this issue by setting class weights and adopting suitable loss functions; however, the imbalance seemed too large to be effectively mitigated. Therefore, we opted to perform downsampling of the majority class (N), initially reducing it by a factor of 6 by randomly discarding samples. Subsequently, we decided to discard signals (N) classified as noise by setting a lower threshold (0.05 dB/Hz) and decreasing the remaining signals by a factor of 4, randomly. We also believe that this approach might help us implicitly for another reason: by training our

models on a dataset of noise-free N signals (with well-defined and clear shapes), we are indirectly teaching the model to classify only clearly distinguishable healthy beats, reducing false positives (sick beats classified as healthy) caused by noise.

3.4 Standardization & Normalization

We tried various approaches to normalize the data, encountering some difficulties, presumably because of the presence of noise (although most of it had been removed).

Initially, we standardized the beats by calculating the mean and variance across all signals from all patients in the training set combined (excluding portions identified as noise by our function). We used the mean and variance values extracted from the training set to normalize the samples in the validation and test sets. However, this approach did not lead to good performance. Then, we tried to min-max normalize each patient's record, how described in [5], but this approach got poor results as well.

Subsequently, we adopted a simpler approach, normalizing the beats between 0 and 1 after extraction. We expected this to degrade performance, as it equalizes the amplitudes of various beats. However, experimentally, the opposite emerged. This might be because the general trend of the signal contains the most relevant discriminants for classification.

4 CLASSIFICATION

Initially, we attempted to train the following models on all three classes for classification, using 3 output neurons, softmax as the activation function, and categorical cross-entropy as the loss function. However, the results were poor, predicting many N classes correctly (which we suspect is more due to the imbalanced nature of the classes), while the others were predicted randomly.

We quickly abandoned this approach and tried training the models initially to distinguish between healthy beats and abnormal ones (combining S and V under a single label). Subsequently, we trained the models to distinguish between S and V (thereby training them exclusively on S and V) and performed fine-tuning for the two models separately.

4.0.1 Optimizer and Loss. As optimizer, we eployed Adam, because of its capability to adapt and dinamically reduce the learning rate when needed. On the other hand, we tried different losses during our trainings. The ones who achieved best results are the Weighed Crossentropy Loss, setting the weights to the inverse class frequency, and the Focal Loss, varying gamma from 1.5 to 2.0.

4.1 Healthy vs Non Healthy

To achieve this, we replaced the last layer with a single output neuron, used the sigmoid activation function, and employed binary cross-entropy as the loss function (although other loss functions specifically designed to address class imbalance were also tested).

This way, the output of the network was a value between 0 and 1, representing the confidence the network had in classifying between the two classes. In this case (distinguishing between healthy and diseased beats), the goal was to minimize the number of false positives (diseased beats classified as healthy). To achieve this, we calculated the ROC curve and determined the optimal threshold parameter that provided the best classification performance (by identifying the threshold corresponding to the maximum True Positive Rate minus False Positive Rate). [see Figure 6]

The best result [see Table 1] was achieved by resnet1 with a threshold of 0.61. In this phase, considering the highly imbalanced classes, we primarily focused on recall and F1 score, with resnet outperforming the others in this regard. Except for biLSTM, all the other networks demonstrated acceptable performance. For the final classification between Healthy and Non-Healthy, resnet was chosen.

4.2 Ventricular vs Sopraventricular

In this phase, the classes were not as imbalanced as before, so we focused less on balancing and more on improving accuracy by fine-tuning and trying different loss functions. However, given the likely challenging nature of the task, achieving high accuracy in the testing phase was challenging. The best result was obtained with ResNet, which achieved an accuracy of 0.69 in the validation phase. Therefore, we decided to use ResNet as the final classifier between S and V.

Model	Accuracy	Precision	Recall	F1 Score
Basic CNN (val)	0.75	0.47	0.89	0.62
Basic CNN (test)	0.69	0.21	0.71	0.32
VGG (val)	0.93	0.78	0.96	0.86
VGG (test)	0.90	0.51	0.71	0.59
ResNet (val)	0.96	0.86	0.96	0.91
ResNet1 (test)	0.93	0.63	0.78	0.69
ResNet2 (test)	0.88	0.45	0.81	0.58
LSTM (val)	0.91	0.75	0.93	0.83
LSTM (test)	0.89	0.48	0.66	0.56
BiLSTM (val)	0.87	0.69	0.74	0.72
BiLSTM (test)	0.82	0.23	0.32	0.27

Table 1: Healthy vs Non-Healthy. Thresholds: CNN: 0.16 VGG: 0.35 ResNet1: 0.63 ResNet2: 0.4 LSTM: 0.17 BiLSTM: 0.24

Model	Accuracy
Basic CNN (val)	0.59
VGG (val)	0.60
ResNet (val)	0.69
LSTM (val)	0.55
BiLSTM (val)	0.58

Table 2: Supraventricular vs Ventricular.

4.3 Model Architectures

Below are all the models used and tested. The results are reported in Tables 1 and 2, as well as in the confusion matrices. The networks were adapted based on the nature of the task, using 1 output neuron in the case of Healthy-NonHealthy classification (to leverage confidence and threshold from the ROC curve) and two neurons for the S, V classification (simply for convenience, as more loss functions are available in the Keras library).

4.3.1 Basic CNN. Firstly, we designed a 1D-CNN. The model consist of multiple layers that progressively extract features from the input data. The architecture includes a series of convolutional layers with ReLU activation, followed by max-pooling to capture essential patterns. At the end of the feature extraction, we employed a global average pooling layer to condense the extracted features, and a dropout layer to mitigate overfitting. The classifier section consists of a dense layer with ReLU activation, followed by the final output layer using a sigmoid.

4.3.2 VGG. Moreover, we adapted two famouse CNN architecture, VGG and ResNet, to our task. VGG (Visual Geometry Group) is a deep convolutional neural network architecture characterized by its uniform structure, comprising multiple stacked convolutional layers with small receptive fields, followed by max-pooling layers, and culminating in fully connected layers for classification.

4.3.3 ResNet. On the other hand, ResNet (Residual Network) is a deep convolutional neural network architecture distinguished by its innovative use of residual connections, or skip connections, that allow the direct flow of information across layers. This design mitigates the vanishing gradient problem and enables the training of extremely deep networks.

4.3.4 LSTM & biLSTM. Furthermore, we Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem, allowing for the effective modeling of long-term dependencies in sequential data.

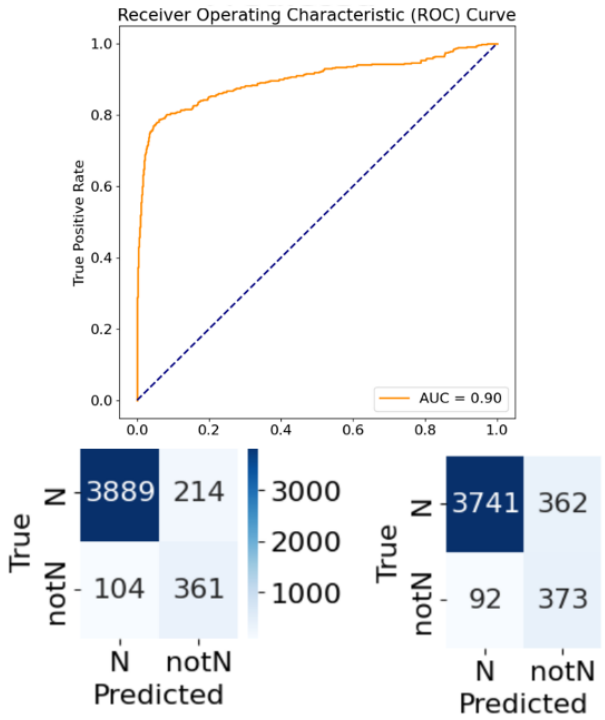

Figure 6: ROC Curve of Resnet predictions. On the left the CM with a threshold of 0.63, on the right 0.4. See how decreasing the threshold decreases the False Positive

LSTMs use specialized memory cells with gating mechanisms, enabling the network to selectively retain or forget information over extended time periods, making them well-suited for time series classification. However, for our task, LSTM did not reach great results, since they were not able to generalize our signals in a proper way.

4.4 Final Model (N,S,V)

Our final model for classifying Photoplethysmogram (PPG) signals is structured to distinguish between normal and abnormal patterns, with a subsequent classification of abnormal beats into Supraventricular (S) and Ventricular (V) categories. So, the model is a series of two different block: firstly, we classify signals in normal (N) and not normal (nonN). Then, we focus on Not Normal beats and classify them more in depth, recognising Supraventricular and Ventricular beats. In both blocks, we employed a ResNet as described above, since it was the model reaching best performances on both classifications. (see Figure 7, 8) The model shows an accuracy of 79%, attributed to the strong class imbalance, and a recall of 59%. The overall results are lower than expected; we tried to minimize false positives as much as possible, yet a significant portion of diseased signals are classified as healthy. One of the main causes could be the presence of residual noise in both the training and testing phases. Further investigations can be carried out to enhance the beat extraction and its normalization. Additionally, data collection could be supplemented with other tools, such as an accelerometer, to more precisely filter out noise caused by movements or vibrations.

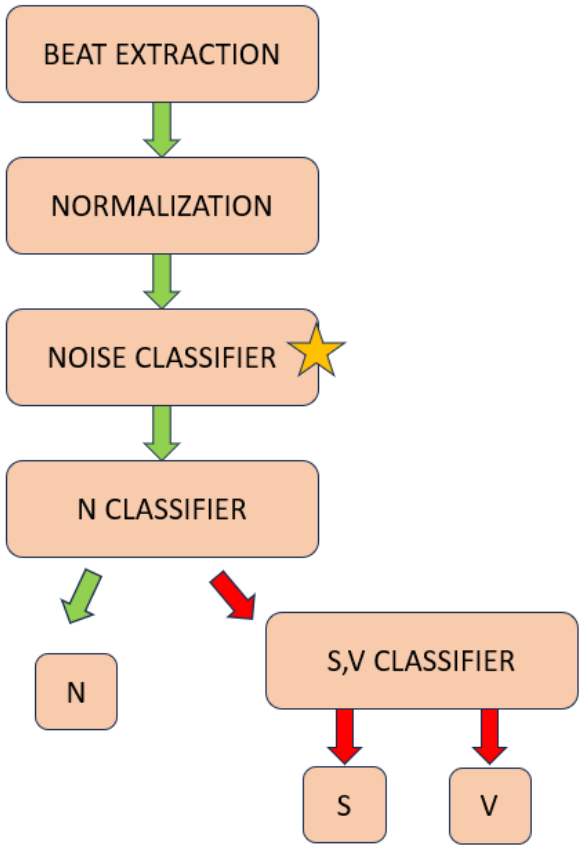

Figure 7: Final Classifier Flow

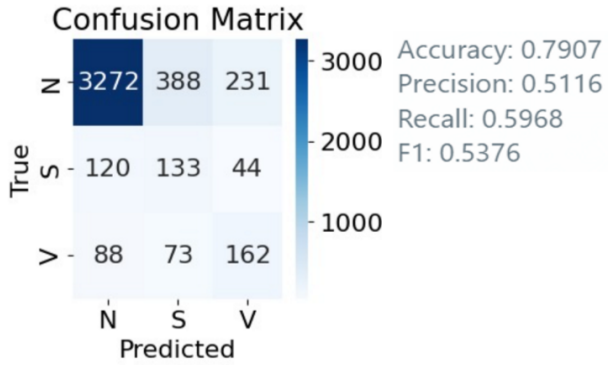

Figure 8: Final Classifier Performance

REFERENCES

[1] Al-Haija et al. Al Fahoum. 2023. Identification of Coronary Artery Diseases Using Photoplethysmography Signals and Practical Feature Selection Process. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9952145/.

[2] Esgalhado et al. Fernandes. 2021. The Application of Deep Learning Algorithms for PPG Signal Processing and Classification. https://www.mdpi.com/2073-431X/10/12/158#.

[3] Liu et al. Li. 2018. Comparison and Noise Suppression of the Transmitted and Reflected Photoplethysmography Signals. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6178150/.

[4] Wang et al. Li. 2020. Classification of Photoplethysmographic Signal Quality with Deep Convolution Neural Networks for Accurate Measurement of Cardiac Stroke Volume. https://www.mdpi.com/2076-3417/10/13/4612.

[5] Zhou et al. Liu. 2022. Multiclass Arrhythmia Detection and Classification From Photoplethysmography Signals Using a Deep Convolutional Neural Network. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9075456/.

[6] Sardana et al. Neha. 2021. Arrhythmia detection and classification using ECG and PPG techniques: a review. ://link.springer.com/article/10.1007/s13246-021-01072-5.