Classification of 12-lead ECGs: The PhysioNet/Computing in Cardiology Challenge 2020

Lucía García-Duarte Sáenz

The objective of this work is to implement algorithm that can, based only on the clinical data provided, automatically identify the cardiac abnormality or abnormalities present in each 12-lead ECG recording.

The analysis was carried out according to the main steps involved in the ECG signal processing stage: data acquisition, pre-processing, feature extraction and classification [1].

To perform heart disease multilabel classification, two different deep learning approaches were considered:

- **ECG signal features with MLP**: several time and frequency domain features were extracted to be fed into a simple dense neural network.
- **1D-CNN as feature extractor**: a one-dimensional convolution neural network was used to extract spatial and temporal dependencies along the ECG segments containing the 12 leads.

1 Data acquisition

Two datasets included in the PhysioNet Challenge were used:

- *PTB-XL Database*: contains 21,837 clinical 12-lead ECGs of 10 second length with a sampling frequency of 500 Hz. A subset of 3,500 trials was selected. Fig. 1 shows a sample of the data.
- *St Petersburg INCART 12-lead Arrhythmia Database*: consists of 74 annotated 30 minutes long recordings extracted from 32 Holter records that contain 12 standard leads, each sampled at 257 Hz. A subset of 20 recordings was selected.

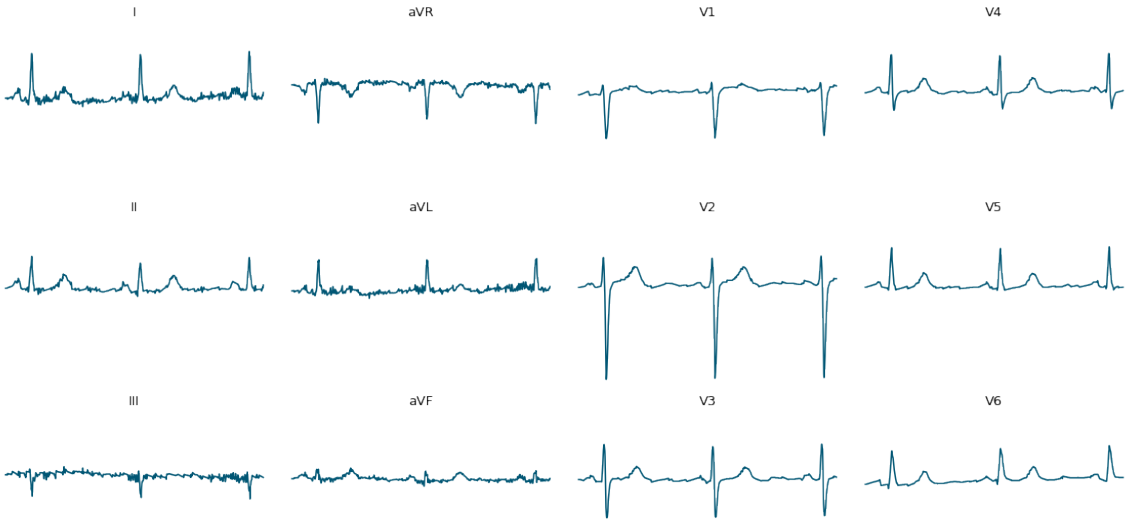

Figure 1: Example of the 12 leads of an ECG.

2 Pre-processing

In order to perform classification, data must be filtered from noise and prepared for processing, for instance by segmenting it into heart beat signals of the same length.

2.1 Sampling frequency matching

For this purpose, the first step consisted on **adjusting the sampling frequency** so that all signals could be further processed simultaneously and all had the same *influence* on classification, meaning that they all can be segmented appropriately. All records with a sampling frequency different from 500 Hz were upsampled using the Fourier method (in which the signal is assumed to be periodic) to match that value.

2.2 Filtering

Then, **data was filtered** to obtain a clean version of the signals more suitable for processing. The following steps were computed:

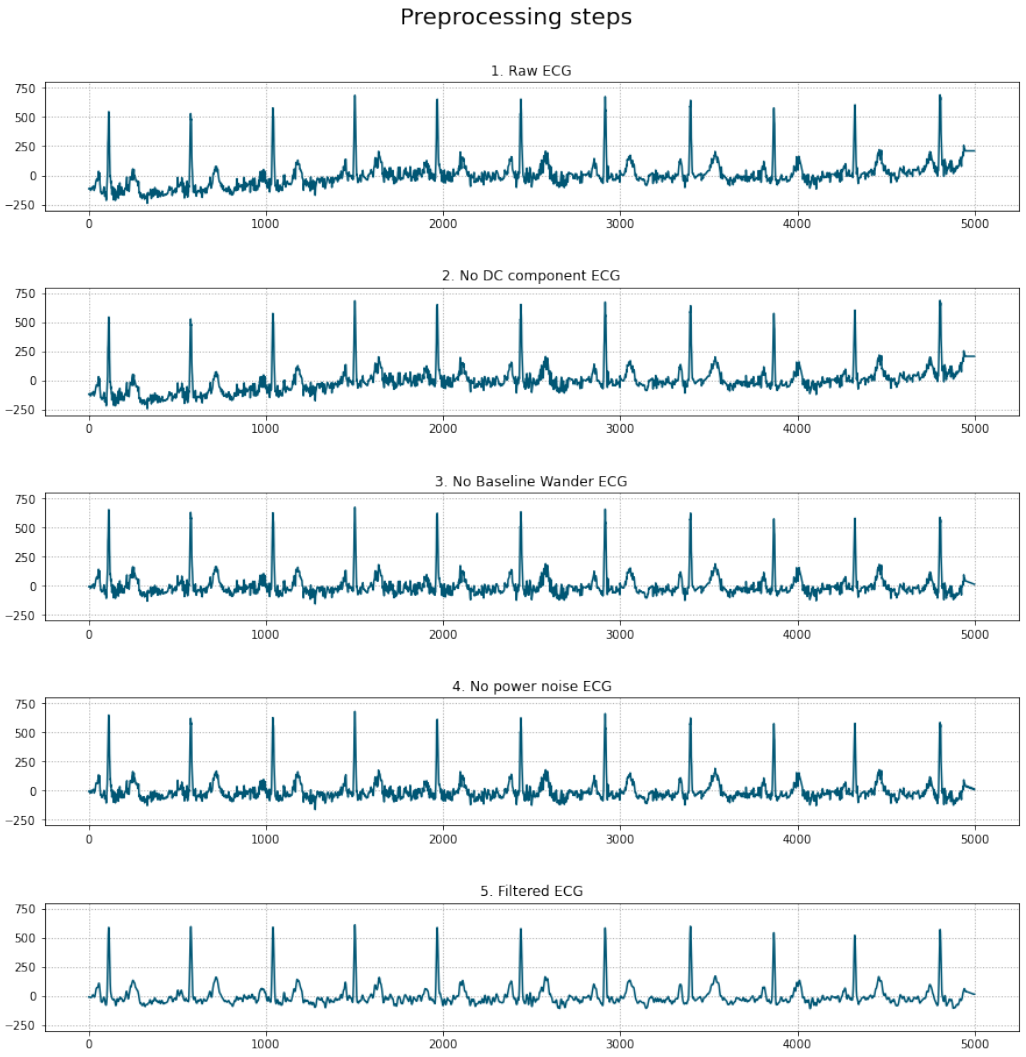

Figure 2: Example of preprocessing steps on a lead I signal.

- *Removal of DC component*: by subtracting the mean of the leads of each signal.
- *Removal of baseline wander*: by applying a high pass filter with a cutoff frequency of 0.5 Hz to remove the extraneous, low-frequency activity registered in the signals.
- *Removal of power line inference*: by applying a notch filter with a cutoff frequency of 50 Hz to remove power noise.
- *Signal smoothing*: by applying a Savitzky–Golay filter with a window length of 25 samples. This type of filter has an important peak preserving property which is very useful in ECG signal analysis.

If we take a look a the frequency domain of both the raw and the preprocessed signals, it can be seen how the signal has been smoothed and no high frequency components are seen. Additionally, the baseline 0.5 Hz and the noise due to power line source have also been removed.

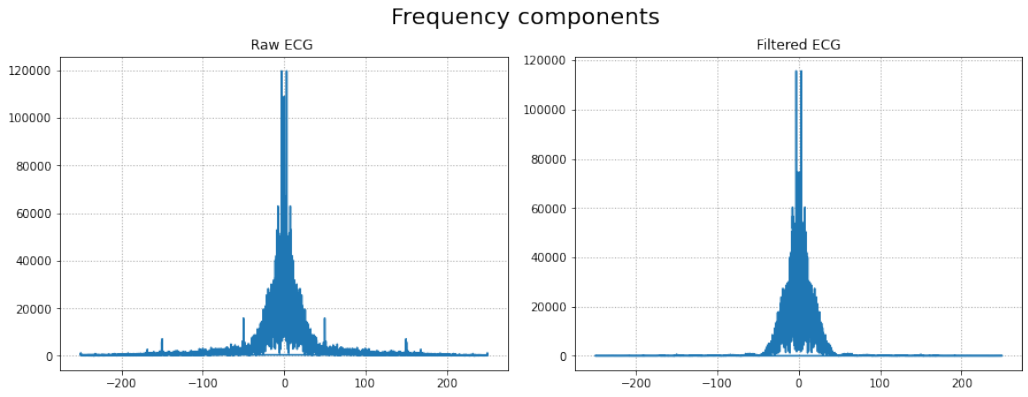

Figure 3: Example of a lead I signal frequency domains before and after preprocessing.

2.3 Segmenting

The final preprocessing stage consists on **segmenting the signals and obtain equal length segments** when needed for further processing:

- *Segmentation*: to segment the signals into heart beats, the *Pan-Tompkins QRS detection algorithm* was first used to extract the R peaks of the signal based on lead I. Then, to separate signals into segments (the 12 leads are processed simultaneously), the samples between the 0.35% of the signal to the left of the R peak and 65% of the signal to the right of the R peak were considered a segment. These values were estimated according to heart beat durations considered in [2].
- *Zero padding and truncation*: to obtain segments of the same length, signals were either chopped o padded with zeros (in a mirrored manner) to a length of 500 samples (1 second duration).

2.4 Feature extraction

For the 1D-CNN, no features were extracted manually from the signals, but rather were automatically extracted by the neurons in the network layers.

On the contrary, for the MLP model, the following features were obtained from each segment:

- *Time-domain* features: mean, standard deviation, median, maximum, minimum, root mean square (RMS), and zero-crossing rate.
- *Frequency-domain* features: spectral energy and wavelet coefficients features (mean, standard deviation, variance, median, maximum, minimum, RMS, zero-crossing rate, and mean-crossing rate).

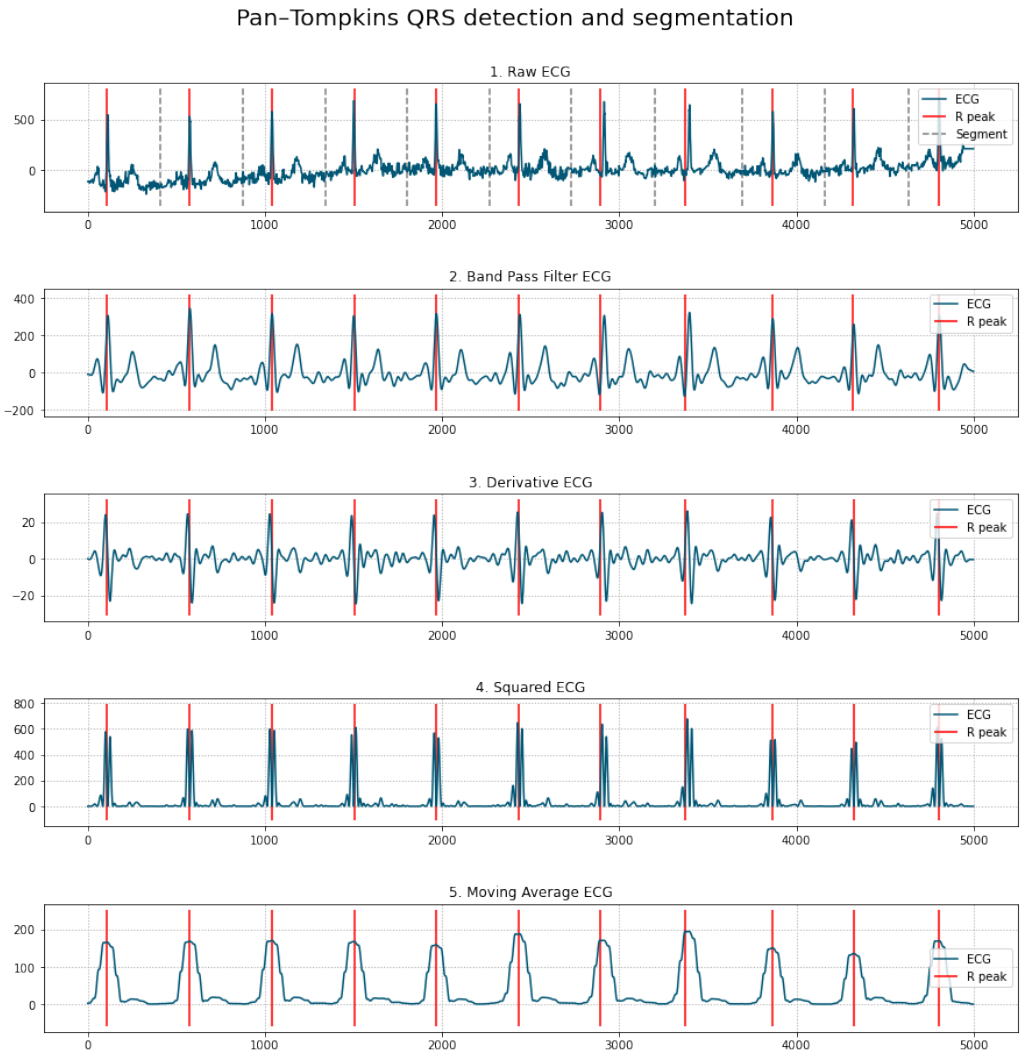

Figure 4: Example of segmentation steps on a lead I signal.

3 Model building

Then, the two models were trained as follows:

3.1 Train-test split

Data segments were divided into train (40%), validation (30%) and test (30%) sets in a stratified way, so that all sets have proportionally the same class distribution.

3.2 Imbalanced data

As observed in 5, class labels present a highly imbalanced distribution, which can lead to overfitting as the model will better learn the most frequent classes. Not only that, but also the lack of data of some specific labels, will result in a model that is not feasible enough. In order to correct the imbalanced problem, a **MLSMOTE** (Multi-Label Synthetic Minority Oversampling Technique) algorithm was implemented over the training set as in here, creating 15000 synthetic samples from the minority classes. Notice that increasing the number of synthetic samples would create an artificial, less *pure* dataset.

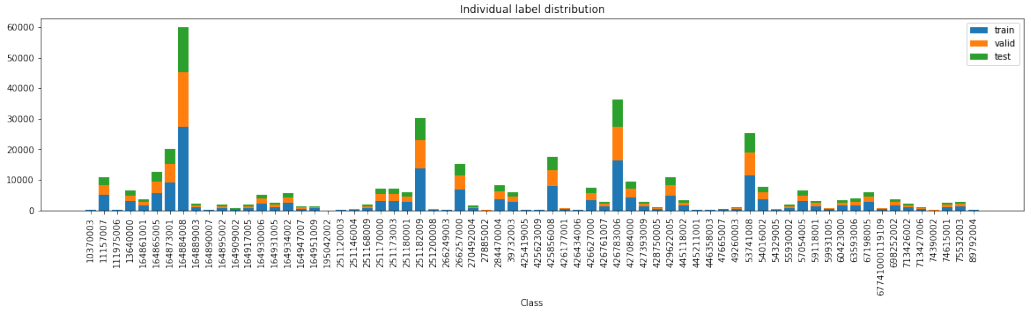

Figure 5: Distribution of the individual labels

Still, data remained imbalanced and many labels lacked presence in the dataset, so further research and analysis should me done to reduce this issue.

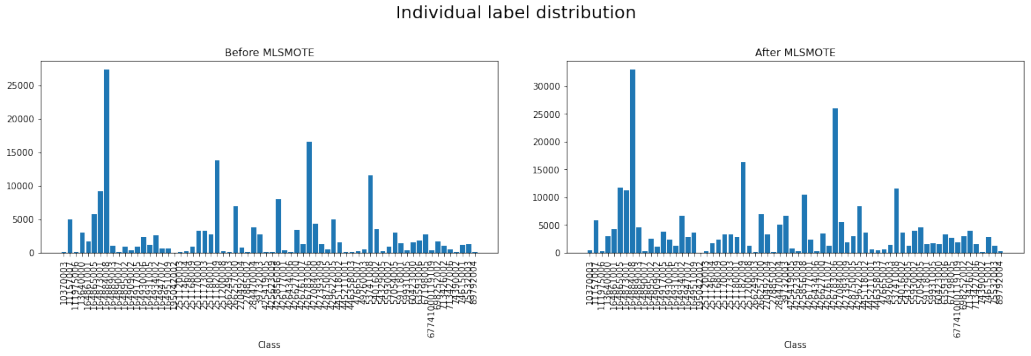

Figure 6: Distribution of the individual labels

3.3 Model training

The two different models were trained accordingly:

- **ECG signal features with MLP**: a dense neural network with two fully connected layers was implemented. Dropout was used for regularization and batch normalization to ensure a faster convergence during training by reducing internal covariance shift. The input data is of shape $[samples, features]$.
- **1D-CNN as feature extractor**: a neural network composed of two 1D-CNN layers followed by a max-pooling layer and a dense layer was implemented. Dropout was included for regularization. The input data is of shape $[samples, segment\_length, lead]$.

Missing values imputation was not performed since there were no missing values in the data. In both cases and since it is a multilabel classification problem, a sigmoid activation function is used in the output layer, and a binary cross-entropy loss is minimized. The heart beats (either in the form of extracted features or filtered signals) were fed into the model in batches of 32 for 20 epochs.

3.4 Model evaluation

After training with the validation and training sets, the test set was used to give an estimation of the models performance. In the table below, it can be seen than the 1D-CNN model yielded much better results that the model trained with time and frequency domain features. In 1D-CNN networks, the kernel slides along the temporal dimension of the data, extracting temporal dependencies, which shows that considering the raw 12 lead signals seems to be a better approach when performing ECG classification.

However, notice that the dataset was still highly imbalanced, so these results are not representative and would require more data to obtain reliable results. More common labels are being predicted correctly favouring the performance of the 1D-CNN model, while the minority classes are wrongly estimated. Additionally, other data as the gender and age of the patients could be included.

	MLP	1D-CNN
F1-score	0.156	0.749
Accuracy	0.028	0.520
Recall	0.168	0.700
Precision	0.352	0.837

References

[1] M. Fikri, I. Soesanti, and H. A. Nugroho, "Ecg signal classification review," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 5, p. 15, Jun. 2021. DOI: 10.22146/ijitee.60295.

[2] S. Saadatnejad, M. Oveisi, and M. Hashemi, "Lstm-based ecg classification for continuous monitoring on personal wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 2, pp. 515–523, 2020. DOI: 10.1109/JBHI.2019.2911367.