POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Beat classification of PPG signals using machine learning and deep learning techniques

Project report for the course Applied AI in Biomedicine

Authors: Robbe De Muynck, Francesco Leni, Samuel Thys

Professor: Prof. Valentina Corino

Academic year: 2023-2024

1. Introduction

Arrhythmia is the most common cardiovascular disease; it is a cardiac conduction disorder manifested as single or multiple irregular heartbeats. An effective way of spotting these abnormalities is offered by the analysis of the Photoplethysmogram (PPG), which is a non-invasive technique able to provide hemodynamic information of the heart by capturing the blood volume changes in the peripheral vessels of the body. Thus, it is possible to identify arrhythmias thanks to the different morphology and duration of these abnormalities with respect to the normal sinus rhythm. The recent trend into wearable devices asks for the continuous monitoring of the patient, hence it is essential to develop tools capable of autonomously detecting and classifying such abnormalities. Many works have investigated Machine Learning (ML) and Deep Learning (DL) algorithms for these purposes. In 2015, Sološenko et al. [8] proposed to extract time and frequency related features for each peak in a sliding window fashion and used an MLP for identifying premature ventricular contractions (PVC) in the signal. In 2022, Liu et al. [4] proposed to use a VGG-16-based architecture to classify a 10 second window of PPG signal into 6 different categories: atrial fibrillation (AF); premature atrial contraction (PAC); premature ventricular contraction (PVC); normal sinus rhythm (SR); supraventricular tachycardia (SVT); ventricular tachycardia (VT). These works prove the feasibility and effectiveness of such methodologies, but the results, especially when dealing with multi-class classification, are still far from optimal. Therefore, the aim of this work is to develop both a binary and a multi-class classifier to discriminate between normal and abnormal peaks and between normal, PVC and PAC peaks. What follows is a comprehensive description of our work, which is structured as an analysis of both ML and DL algorithms preceded by a common preliminary data analysis and preprocessing step. Results will be provided for the best obtained model inside both frameworks, and conclusions will be drawn to determine the strengths and weaknesses of both of these paradigms.

2. Materials and methods

2.1. Data Analysis

The dataset that was provided is a set of files containing information about PPG recordings for 121 different patients; for each patient, the raw PPG recording, index (or position) of every peak, and label of every peak are provided as

3 separate .mat files. In this way, the peaks are labeled as being normal (N), premature atrial (S, for supraventricular) or premature ventricular (V) contractions. Additionally, for each patient, the sampling frequency is provided in the name of the corresponding files (S001_128.mat has a sampling frequency of 128 Hz, for example), which needs to be taken into account when considering any temporal or frequency-related information.

The amount of peaks for each class throughout the whole dataset was inspected, as this needs to be taken into consideration when developing a ML or DL model that can discriminate different classes in a desirable way. For the task at hand, there are two ways of viewing this potential class imbalance as there are technically two tasks to consider: the binary classification task needs to differentiate normal (N) beats from abnormal ones (including both S and V beats), while the multi-class classification task needs to differentiate all three possible classes. Significant class imbalances were found for both taks, as seen in Figure 6, although within the abnormal peaks, the S and V beats seem to be present in comparable amounts: 9691 S beats and 7994 V beats were counted, while 226448 N beats are present in the dataset.
Plotting the raw PPG signals over time reveals the presence of a substantial amount of high frequency noise as well as motion artifacts, as seen in Figure 1 (top). These will be taken into account in following sections.

2.2. Common Preprocessing

Some preprocessing steps, which both frameworks have in common, were performed on the provided data to ensure consistency in the treatment of the data across the development of all ML and DL models that were tested. To facilitate this process, efforts were made to make optimal usage of a Python toolbox called pyPPG, developed by Goda et al. in 2023 [2], which provides implementations of algorithms specifically made for the loading and filtering of raw PPG signals. Specifically, pyPPG's implementation of forward and backward filtering of the signal using Chebyshev (type II) filtering was used to take advantage of Chebyshev filters' sharp roll-off to the stopband while preventing zero-phase distortion by applying the filter in both directions. The values used for low frequency and high frequency cutoffs were 0.5 Hz and 4.3 Hz respectively, as these are not only the default cutoff frequencies provided by pyPPG but are also in a range of values close to the normal pulse frequency of PPG signals [5].

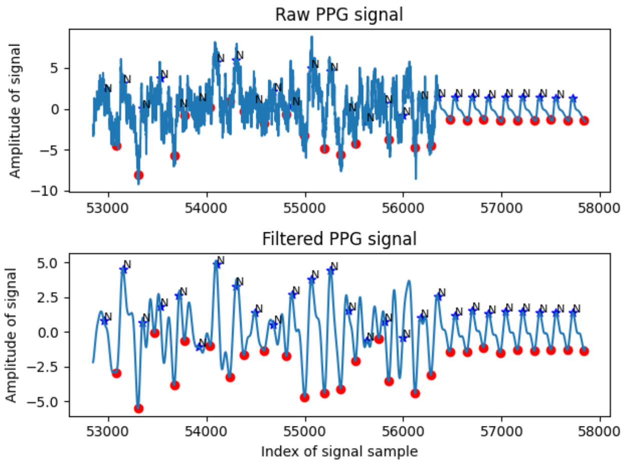

Figure 1: *Comparison of a raw and filtered PPG signal. The red dots represent the onsets extracted using the pyPPG library, hence the portion of the signals between 2 such points is a crop. This figure also shows the effect of motion artifacts to the signal.*

To see the effect of this filtering in a more concrete way, the spectral density of the signal can be analysed both before and after filtering through the use of periodograms. In Figure 2, it can be seen that the bulk of the spectral density is spread out over the lower (0-1.5 Hz) frequency range, and that the spectral density is also somewhat spread out over higher frequency ranges (3 Hz and up).

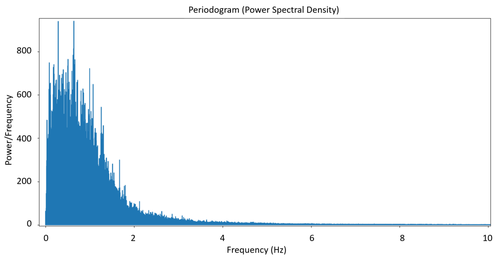

Figure 2: *Periodogram of a raw PPG signal*

After filtering, it can be seen in Figure 3 that the higher and lower frequencies were successfully eliminated and that the desired frequency range is kept almost intact.

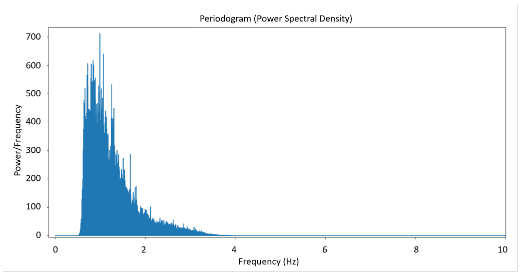

Figure 3: *Periodogram of a filtered PPG signal*

It has to be noted that motion artifacts, which can quite clearly be seen in Figure 1 up until about the midway point between index 56000 and 57000 of the example signal that is plotted, are not adequately removed by the filtering that is applied. Filtering out this noise would result in the loss of crucial information in the PPG signal, so it was decided that these artifacts wouldn't be dealt with by suppressing certain frequencies but by performing some outlier removal techniques within the ML and DL frameworks (see Section 2.3.1 and Section 2.4.1).

It was decided that, for both the ML and the DL frameworks, models would be trained and tested on 'crops' of PPG signals, each containing exactly one peak that is labeled (or needs to be labeled at test time) as 'N', 'S', or 'V'. To do so, 'onsets' were extracted using pyPPG for each patient's full PPG signal: For each pulse that is detected by pyPPG, the onset of the peak is computed. These onsets were then used to define the crops: the onset of a pulse was then taken as the start of a crop, while the onset of the next pulse was used to define the endpoint of that crop and the start of the next. However, some problems were encountered during this cropping, as pyPPG failed to detect each peak that needed to be labeled or even detected peaks where none is present, resulting in the absence of an onset or the presence of more than one onset between two successive peaks. To correct this, an onset was added at the midway point between two peaks when no onset was present, while the onset closest to the second peak was considered as the dividing point between two crops when more than one onset was present. Applying this algorithm to the entirety of the provided dataset resulted in the creation of 244,133 crops, each containing a labeled peak.

2.3. Machine Learning

A necessary step in machine learning is defining a set of features that defines each sample that is used for training and testing. Many features can be implemented purely by computing a set of values using the information present inside each individual crop. These so-called 'intra-crop' features are defined as follows: the duration or length of the crop (crop_duration), the time passed between the onset of the peak and the actual peak (t_peak), the mean value of the crop (mean), the median value of the crop (median), the standard deviation of the amplitudes found within the crop (std), the trimmed variance of the crop (tvar), the skew and kurtosis of the crop (skew, kurt), the area under the curve (auc, using Simpson's rule for numeral integration), the amplitude difference between the minimum and maximum values of the crop (peak_amplitude), the pulse width at half maximum of the peak (pulse_width), the symmetry index (symmetry, found by computing the ratio between the mean value of the first half of the crop and second half of the crop), the spectral entropy (spectral_entropy), the average energy (average_energy), the root mean square of successive differences (rmssd), and the standard deviation of the spectrogram that can be computed for the crop (std_spectrogram). Moreover, as peaks may influence the behavior of succeeding peaks, two features were added to take this temporal behavior into account: the peak-to-peak time (PTP) and the ratio of the amount of beats found to be abnormal (having a 'S' or 'V' label) to the amount of beats found to be normal ('N') in the 20 seconds preceding the crop (A_to_N_ratio). Table 1 shows an overview of all features that were used. The resulting dataset, now consisting of a set of features defining each crop grouped by patient, was split into a training set (containing the crops from 70% of the patients), a validation set (15%), and a test set (15%).

Feature number	Feature name
1	crop_duration
2	t_peak
3	mean
4	median
5	std
6	tvar
7	skew
8	kurt
9	auc
10	peak_amplitude
11	pulse_width
12	symmetry
13	spectral_entropy
14	average_energy
15	rmssd
16	std_spectrogram
17	PTP
18	A_to_N_ratio

Table 1: *List of all features implemented for ML*

2.3.1 Outlier Removal and Subsampling

To account for the fact that outliers may still be present, due to motion artifacts (as mentioned in Section 2.1 or other sources of error, an outlier removal procedure was performed. To do so, a scaler was fitted on the **training** set and used to compute the z-score of each feature across all samples in both the **training** and **validation** sets. Crops containing a feature with a z-score higher than 3 were then removed only from these sets, since we don't want the outliers to influence the learning process. A visualisation of the distribution of the outliers found per feature can be found in Figure 4, while the resulting amount of crops of each class in each set can be found in Table 2. Before removing these crops, the dataset was inspected for any missing values (or 'nan's), but none were found.

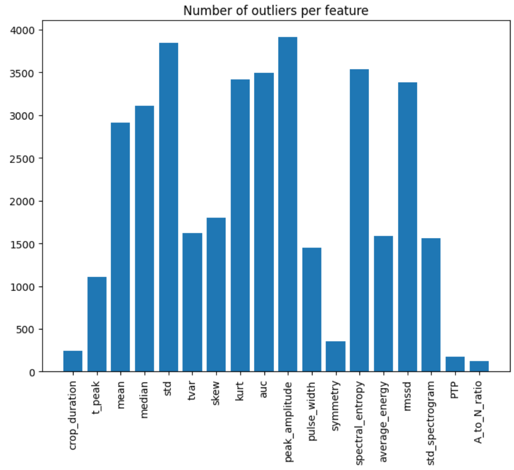

Figure 4: *Number of outliers for each feature. Values exceeding a z-score of 3 are considered outliers.*

	N	V	S
Training	148044	4041	5397
Validation	26276	711	543
Testing	35199	2355	1146

Table 2: *Distribution of samples across training, validation and test sets.*

The class imbalance problem mentioned in Section 2.1, which would cause an ML model to mainly learn from 'normal' crops as there is an overabundance of such crops in the dataset, can be dealt with by means of subsampling. To do so, for the binary classification task, the amount of samples found within the class containing the least amount samples was used to subsample the other classes, except for the 'normal' class which was subsampled twice that amount, resulting in a distribution of 50% N crops and 50% A (abnormal), or 50% N, 25% S, and 25% V crops. For the multi-class classification task, a similar procedure was performed, except that the 'normal' class was now subsampled to obtain the same amount of samples as the amount of samples present in the smallest class. This was done for the training and validation sets as doing this for the test set is trivial. The resulting amounts of crops of each class for the training and validation sets can be found in Table 3.

	N	V	S
Training (binary)	8082	4041	4041
Validation (binary)	1086	543	543
Training (multi-class)	4041	4041	4041
Validation (multi-class)	543	543	543

Table 3: *Distribution of samples across training, validation and test sets: after subsampling majority classes.*

2.3.2 Data standardization

The dataset was then standardized by subtracting the values of all features with the mean value of each feature in the **training set** and scaling them to unit variance by dividing them by the standard deviation (also of each feature in the **training set**). This step is necessary since ML models tend to behave better when the data looks more normally distributed [7]. The scaler, with values computed from the training set, was also applied to the test and validation sets to ensure consistency across the development of the models.

2.3.3 Feature correlation Analysis

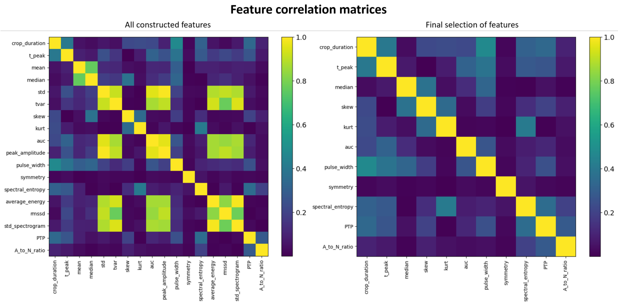

Figure 5: *Correlation of features (left: all features, right: selection of features) generated on the training dataset. A correlation threshold of 0.75 was used to remove correlated features.*

The dataset, now cleaned after removal of outliers, subsampling and standardization, was then analysed for correlation between each feature, to see whether or not some features contain redundant information already captured by others; removing those could potentially significantly reduce the size of the dataset, making the training of models less computationally expensive. By inspecting all correlations (Figure 5, left), it was observed that some features showed higher correlation with each other (above 0.75). Of each pair of these highly correlated features, one feature was dropped, resulting in the removal of 7 features: standard deviation (std), standard deviation of the spectrogram (std_spectrogram), trimmed variance (tvar), peak amplitude (peak_amplitude), root mean square (rmssd), mean (mean), and average energy (average_energy). Table 4 shows the remaining features, while Figure 5 (right) shows the correlation between each of these remaining features.

Feature number	Feature name
1	crop_duration
2	t_peak
4	median
7	skew
8	kurt
9	auc
11	pulse_width
12	symmetry
13	spectral_entropy
17	PTP
18	A_to_N_ratio

Table 4: *List of all features for ML after removing correlated features*

2.3.4 Architecture

The architecture that was chosen for ML was a logistic regression model, in which every feature of the input sample is associated with a weight; the weighted linear combination of the input features is then fed through an activation function, after which a decision boundary will determine whether the sample belongs to one or the other class in the case of binary classification tasks. For multi-class classification tasks, all classes are simultaneously considered by having a set of weights for each class; a softmax activation converts the values obtained for the

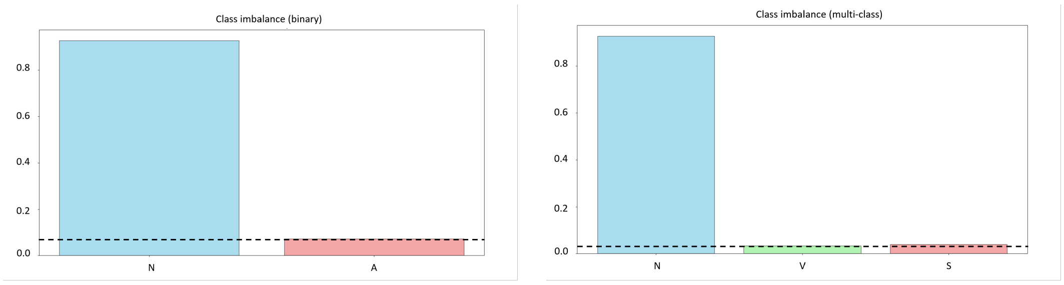

Figure 6: *Representation of the dataset imbalance both binary and multi labelled. The dashed line represents the number of samples inside each class that were used for training. N: Normal; S: premature atrial contraction; V: premature ventricular contraction; A: S+V.*

sample after weighting with each set of weights individually into probabilities of which the highest is chosen as the predicted class. The main reason for choosing this architecture is the fact that it can handle large amounts of data (which is the case for the problem at hand, thanks to the simplicity of the algorithm.

2.4. Deep Learning

DL can be seen as an extension of ML where the feature extraction routine is also addressed by the model, hence DL allows one to extract complex and semantic information of the input to generate a rich description of it in the feature space. An analysis of different models' architectures, which will be explored in Section 3.3, was performed to try to identify the most suitable one for both the binary and the multi-class classification problem. What follows is the description of the 2 models that performed the best for each task.

2.4.1 Outlier Removal

As mentioned in Section 2.1, the presence of motion artifacts was found to be an issue as it was dramatically spoiling the performance of any trained model. Hence, a strategy to selectively remove them had to be taken into account.

$$\begin{cases} \text{Th\_low} = \text{Q1} - \text{IQR} & \text{(1a)} \\ \text{Th\_up} = \text{Q3} + \text{IQR} & \text{(1b)} \end{cases}$$

The most straightforward way of removing them is to define an amplitude threshold for both an upper and lower limit above which we can safely consider the signal to be too distorted to be correctly classified. Crops that were exceeding this limit were discarded, obtaining a final dataset composed of 152337 crops.

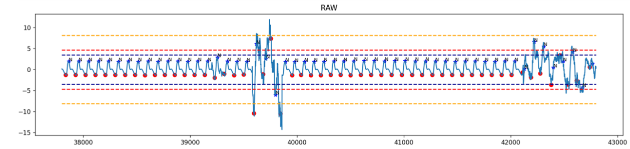

Figure 7: *Visualization of the different analyzed thresholds. The yellow lines are defined using 3 times the IQR in the calculation of the thresholds, while the red ones use 1.5 times the IQR and the blue ones use 1 time the IQR and are also the used ones. It is clearly visible that the more restrictive ones allows for a better rejection of such distortions.*

The threshold value was tuned empirically based on the obtained performances, on the amount of removed data, and on visual analysis. In the end, a threshold equal to the first quartile (Q1) minus the Inter Quartile Range (IQR) was set for the lower bound (eq. 1a), while the third quartile (Q3) plus IQR was defined as the upper limit (eq. 1b).

2.4.2 Training Strategy

The models were trained to classify each crop individually regardless of any relationship with the surrounding ones. Batches of 256 crops were created and the model was trained over them for 50 epochs. The data used for training are the unfiltered crops, as we found no empirical evidence of any improvement brought by the filtering of them that was mentioned in Section 2.2. To train the investigated models, a common pipeline has been followed to ensure comparability of the results. First, the data was split into a train and test test from 85% and 15% of the dataset respectively, then a portion of the training set of equal size to the test set was kept for validation; the dataset was stratified during the division to keep the original class distribution. To compensate for the extreme imbalance of the data (*figure* ??) both weighted loss and sub-sampling were investigated. In the end, a kind of 'adaptive' sub-sampling was implemented to make full use of the entire amount of data while ensuring the perfect balancing of classes during training. For instance, the number of samples from the less frequent class was used as the target number of crops for sub-sampling the majority ones, but the data were shuffled and resampled before every new epoch to ensure a new subset of them to be used during training. Min-max scaling was then applied to the remaining crops to normalize the amplitude inside them between 0 and 1 within the considered interval. The models were trained using the AdamW optimizer to minimize CrossEntropyLoss. The initial learning rate was set to 4e-4 and a cosine decay scheduler was used to decrease it smoothly towards 4e-7. To account for overfitting, Weight Decay regularisation was implemented with a 0.05 factor and Early Stopping was used, which monitors the validation loss and always retains the best model in term of loss. Label smoothing and dropout were also investigated but no real improvements could be achieved and hence were not used for the final model.

2.4.3 Proposed Models

The proposed models are both built upon a 1D Convolutional Neural Network (CNN) to exploit the inherit scaling and hierarchical representation ability of convolutions that allow for a rich and semantically meaningful representation of the input. The binary and the multi-class classification task were best accomplished by two different architectures: a ResNet-like one for the binary task and a DarkNetCSP-like for the multi-class one.

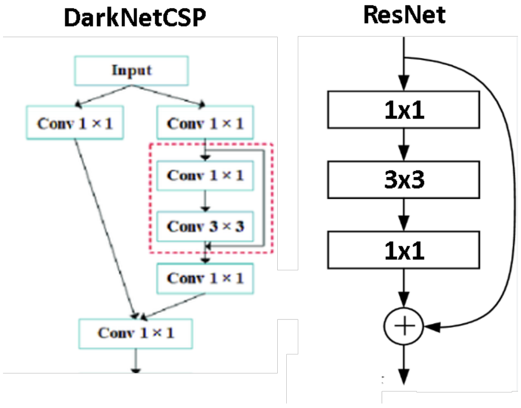

Figure 8: *Comparison between a DarkNetCSP and a ResNet block.*

Both the models were composed of 4 stages with a stage ratio of 3, 6, 9 and 3. Between each stage the time dimension of the crops was downsampled by a factor 2 thanks to strided convolutions. Global Average Pooling (GAP) was used at the end of the feature extractor to obtain a fixed size feature vector to be fed to the classifier. The classifier is a simple Feed-Forward Neural Network outputting 2 and 3 values for the binary and multi-class task respectively. These outputs are passed to a softmax layer to obtain the final probabilities and, consequently, the final prediction.

3. Results

3.1. Evaluation Metrics

To evaluate the performances of the proposed models a common set of metrics has been use to ensure comparability and significance among the different results. Recall (R) (eq. (2a)) was taken as the reference metric for our analysis since, as it is defined as the ratio between the True Positive (TP) and the TP plus the False Negative (TN), it defines the ability of the model to correctly classify a sample of a specific class as belonging to that class. This ability is crucial when developing tools that need to be more sensible than accurate. As arrhythmias can be associated with a higher risk of stroke and hearth attack [9], an early detection of it can be vital.

$$
\begin{cases}
R = \dfrac{TP}{TP + FN} & \text{(2a)} \\
P = \dfrac{TP}{TP + FP} & \text{(2b)} \\
A = \dfrac{TP + TN}{TP+TN+FP+FN} & \text{(2c)} \\
F1 = \dfrac{2}{\frac{1}{P} + \frac{1}{R}} & \text{(2d)}
\end{cases}
$$

Along with this metric, Precision (P) (eq. (2b)), Accuracy (A) (eq. (2c)) and F1 score (eq. (2d)) were taken into account to more completely inspect the performances of the models. Precision is defined as the ratio between the TP and the TP plus the False Positive (FP) and is, hence, a descriptor of how many of the positive classified samples were actually positives. Accuracy is defined as the ratio between the correctly predicted samples over the entire amount of samples of any class, meaning it is a general indicator of the overall quality of the prediction. The F1 score is defined as 2 over the sum of the inverse of Recall plus the inverse of Precision and can be interpreted as an overall measure of the quality of the prediction within each class. Since these metrics, except for accuracy, are primarily intended to be used for binary tasks, it was decided that, to analyze the result for the multiclass problem more adequately, unique descriptors would be created for the models. This is done by combining a metric (M), computed for each of the indiviual non-normal classes (V and S), into a single one by weighting them according to their relative frequency (eq. (3a)).

$$
\begin{cases} M = \dfrac{V}{V+S} \times M(V) + \dfrac{S}{V+S} \times M(s) \end{cases} \quad \text{(3a)}
$$

It is worth noting to say that in this analysis a Sample is considered to be Positive if it belongs to that specific class, while a Negative sample is simply one not belonging to it. Thus, the meaning of every metric depends on the class on which it is computed and. For instance, in the binary case, what is commonly referred to as Sensitivity will be the Recall computed on class 1, while Specificity will be the Recall computed on class 0.

3.2. Machine Learning

3.2.1 Binary Classification

The results, i.e. the metrics and confidence scores for different implemented architectures based on the features selected in Section 2.3.3 can be found in Table 5. The proposed model, i.e. the Logistic Regression model, obtained the highest Recall (80.6%) for class 1 (arrhythmias). The model resulted in 96.9% Recall for class 0 (normals). Precisions of 98.1% and 72.1%, and F1 scores of 97.5% and 76.1% are obtained for class 0 and class 1 respectively, resulting in a overall accuracy of 95.4%. The model predictions are supported by confidence scores of 94.1% and 83.4% for class 0 and class 1, with an overall confidence score of 93.1%.

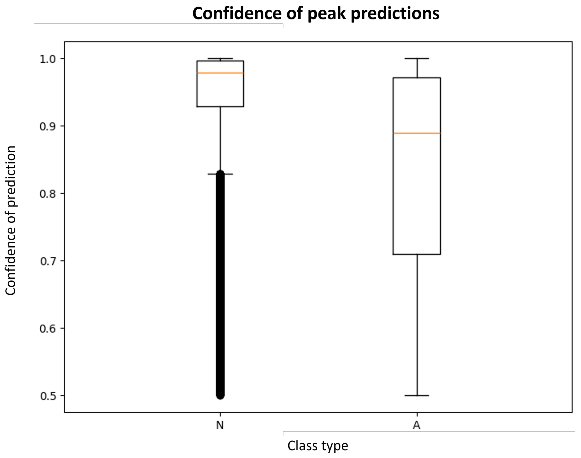

Figure 9: *Confidence score of each class, extracted from the Logistic Regression model on the binary classification task.*

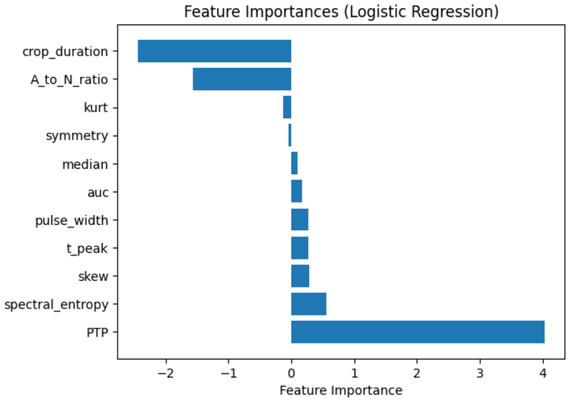

Figure 10: *Feature importances extracted from the proposed Logistic Regression model for the binary classification task.*

	R_0	P_0	$F1_0$	R_1	P_1	$F1_1$	A	Conf	Conf_0	Conf_1
Logistic Regr.	96.9	**98.1**	97.5	**80.6**	72.1	**76.1**	95.4	**93.1**	**94.1**	**83.4**
Random Forest	97.9	97.2	**97.6**	71.5	77.4	74.3	**95.5**	92.3	93.6	77.7
SVM (rbf kernel)	97.9	94.1	96.0	79.9	57.4	66.8	92.8	91.9	94.1	77.0

Table 5: *Comparison between the proposed Logistic Regression model and other investigated ML architectures on binary classification. Classes 0 and 1 correspond to N and A (V+S).*

	R_0	P_0	$F1_0$	R_{1-2}	P_{1-2}	$F1_{1-2}$	A	Conf	Conf_0	Conf_1	Conf_2
Logistic Regr.	96.4	**98.1**	97.3	**42.6**	51.8	42.1	91.6	**86.4**	90.5	49.9	55.6
Random Forest	**98.0**	97.1	**97.5**	42.0	**53.4**	**45.6**	**92.9**	86.3	89.2	**51.2**	54.4
SVM (rbf kernel)	93.0	98.0	95.5	41.7	40.5	36.0	88.4	85.2	**90.6**	49.9	**55.9**

Table 6: *Comparison between the proposed Logistic Regression model and other investigated ML architectures on multi-class classification. Classes 0, 1 and 2 correspond to N, S and V.*

The coefficients of the Logistic Regression model for each feature are plotted in Figure 10. Each coefficient represents the change in the log-odds of the labelling of a peak as normal, associated with a one-unit change in the corresponding feature, while holding other features constant.

3.2.2 Multi-class Classification

The metrics and confidence scores for different implemented architectures based on the features selected in Section 2.3.3 can be found in Table 6. The proposed model, i.e. the Logistic Regression model, obtained the highest weighted Recall (42.6%) for class 1 and 2 (arrhythmias, weighted as described in Section 3.1). The model resulted in 96.4% Recall for class 0 (normals). A precision of 98.1% and F1 score of 97.3% were obtained for class 0, while a precision of 51.8% and an F1 score of 42.1% are obtained for class 1 and class 2, again weighted according to eq. 3a. The Logistic Regression resulted in an overall accuracy of 91.6%. The model predictions are supported by confidence scores of 90.5%, 49.9% and 55.6% for N, S and V predictions respectively, with an overall confidence score of 86.4%.

3.3. Deep Learning

The test set used for this analysis has been cleaned from outliers using the same procedures mentioned in Section 2.4.1, but it has not been stratified to resemble as closely as possible a realistic scenario where normal peaks are typically far more present than abnormal ones, hence it can be said that these results can be a reliable descriptor of the proposed models' performances. The proposed models, along with all other models that were investigated, have been tested on the same test. Hence, a comparison between them will be portrayed.

3.3.1 Binary Classification

As shown in Table 7 the proposed model obtained the highest Recall (95.1%) for class 1 (arrhythmias) and a Recall of 88.3% for class 0 (normals), while it obtains a Precision of 39.8% for class 1 and 99.6 % for class 0. The F1 score inside class 1 is 56.1% and 93.6% for class 0. An overall accuracy of 91.7% was obtained. The prediction was supported by a mean confidence value of 90.8%, but, after dividing into single categories, an 84.5% mean confidence is obtained inside class 1 and a 90.8% mean confidence is obtained inside class 0. The confidence score (Figure 11) is the predicted probability of the input peak belonging to the predicted category, hence it is the actual output of the model after softmax.

3.3.2 Multi-class Classification

As shown in Table 8 the proposed model obtained the highest Recall (71.2%) for class 1 and 2 (arrhythmias, weighted as described in Section 3.1) and a Recall of 88.3% for class 0 (normals), while it obtains a Precision of 36.3% for

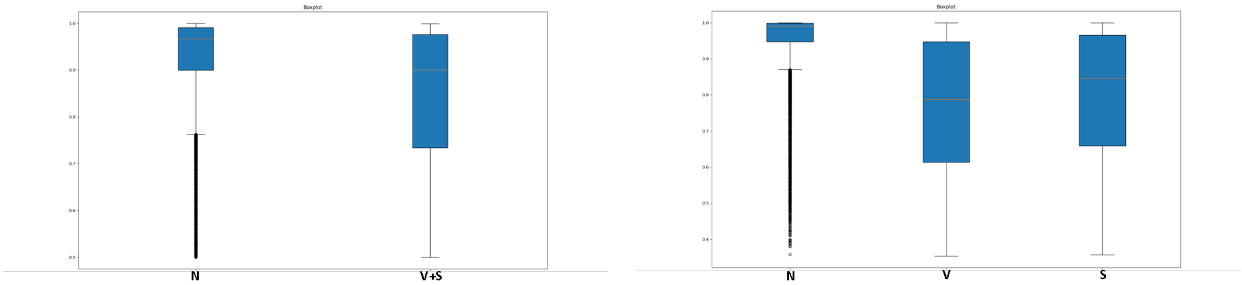

Figure 11: *Comparison of confidence scores per class of the proposed model for both of the tasks: Resnet on the left for the binary problem and DarkNetCSP on the right for multi-target classification.*

	R_0	P_0	$F1_0$	R_1	P_1	$F1_1$	A	Conf	Conf_0	Conf_1
ResNet	88.3	**99.6**	93.6	**95.1**	39.8	56.1	91.7	90.8	92.1	84.5
DarkNetCSP	89.1	99.5	94.0	94.4	41.3	57.4	91.8	91.3	92,6	**85.0**
ConvNeXtSAM	**91.6**	99.4	**95.3**	92.8	**47.3**	**62.6**	**92.2**	**92.0**	**93.3**	84.3
biLSTM	90.2	99.0	94.4	89.2	42.6	57.6	89.8	90.0	91.1	83.6

Table 7: *Comparison between the proposed ResNet model and the other tested architectures. As mentioned, the proposed model shows the highest Recall for class 1 (Sensitivity) while having the lowest Recall for class 0 (Specificity)*

class 1 and 2 and 99.3 % for class 0. The F1 score inside class 1 and 2 is 48.1% and 93.4% for class 0. An overall accuracy of 77.8% was obtained. The prediction was supported by a mean confidence value of 94.5%, but, after dividing into single categories, an 80.2% mean confidence is obtained inside class 2, a 76.9% mean confidence is obtained inside class 1, and a 90.8% mean confidence is obtained inside class 0. A closer look into the performances of the proposed model for the individual classes can be seen in Table 10. The confidence score (Figure 11) is computed in the exact same way as described in Section 3.3.1.

4. Discussion

4.1. Machine Learning

As can be seen in Tables 5 and 6, the application of ML algorithms on a set of conscientiously constructed features, as described in Section 2.3, yields a valid potential in tackling both binary and multi-class classification problems in the domain of PPG-analysis.

When extracting the feature's importance of the proposed Logistic Regression model, one can identify that both PTP and A_to_N_ratio proved to be impactful features for the model's prediction. These features could be labelled as 'inter-crop features' and their relevance confirms the initial assumption that capturing information on the relation between peaks provides useful information with respect to the class they belong to (Section 2.3). When earlier peaks were labelled as arrhythmias (A), following peaks are more prone to get labelled as arrhythmias as well (A_to_N_ratio. This behaviour can be perceived by visual inspection of the PPG recordings, accompanied by the systolic peak labels. Additionally, the spectral_entropy was found to be a predictive feature as well, indicating that features incorporating frequency content could potentially harbour discriminative ability when assessing arrhythmias, alongside temporal features. Temporal features include for example crop_duration as longer crop lengths tend to indicate arrhythmias.

A discrepancy in the model performance could be observed: the model's prediction strongly relies on the past peak labels, as it computes the A_to_N_ratio metric based on the peak labels in the last 20 seconds. During model construction, these peak labels are known, while during testing they are unknown.

During testing, the labels of past peaks should

	R_0	P_0	$F1_0$	R_{1-2}	P_{1-2}	$F1_{1-2}$	A	Conf	Conf$_0$	Conf$_1$	Conf$_2$
DarkNetCSP	88.3	**99.3**	93.4	**71.2**	36.3	48.1	**77.8**	**92.2**	**94.5**	**76.9**	**80.2**
ResNet	**91.9**	99.1	**95.3**	69.6	37.0	48.1	76.7	88.6	91,6	65.6	71,9
ConvNeXtSAM	89.7	97.7	93.5	71,0	**37.6**	**49.1**	77.8	90.7	93.7	70.1	74.1
biLSTM	84.1	98.6	90.8	49.8	18.5	25.0	61.5	82.2	87.4	35.4	52.2

Table 8: *Comparison between the proposed DarkNetCSP model and the other tested architectures. As mentioned, the proposed model shows the highest Recall for classes 1 and 2 weighted (Sensitivity), and also shows the highest confidence score*

	R_0	P_0	$F1_0$	R_1	P_1	$F1_1$	A	Conf	Conf$_0$	Conf$_1$
Known previous peaks	93.0	**99.3**	96.1	**93.8**	57.3	71.2	93.1	91.8	92.6	**87.0**
'online' testing	**96.9**	98.1	**97.5**	80.6	**72.1**	**76.1**	**95.4**	**93.1**	**94.1**	83.4

Table 9: *Comparison between the proposed Logistic Regression on binary classification of the test dataset: 'online' testing versus case where previous peaks are known. Classes 0 and 1 correspond to N and A (V+S).*

be equal to the model prediction for these peaks. However, utilizing the ground truth annotations for this past data when classifying the currently considered peak cannot be considered equally meaningful, as the ground truth will not be provided in real-life scenarios. To provide an insight into the performance of a correctly inferred test set, a recursive implementation of the feature extraction was assessed, which treated the test data labels as 'unseen'. In Table 9, one can clearly distinguish a performance drop in terms of the Recall of arrhythmias: 80.6% for 'online' testing versus 93.8% for known previous peaks. Additionally, the confidence of predictions for the arrhythmia class drops from 87.0% to 83.4%.

4.2. Deep Learning

The results shown in Table 7 and 8 shows the potentialities of DL tools for analysing and classifying peaks extracted from a PPG signnal, while it is also clear that there is still room for improvements. As mentioned, both the proposed models were chosen due to their highest recall, but this was, in both cases, accompanied by a very low precision. This result can mainly be attributed to the non-stratification of the test set. As many more normal peaks were present, and since the model was already more prone to predicting 1 instead of 0, it seems reasonable to obtain such a result. In fact, when looking at the results computed on a stratified dataset, the precision reaches 89.0% while the recall keeps a value of 95.0%.

	N	V	S
Recall	88.3	67.3	74.5
Precision	99.3	34.3	38.1
F1	90.1	45.4	50.4

Table 10: *Comparison of the metrics inside the individual classes for the DarkNetCSP model proposed for the multi class problem*

This result may also suggest that the 'adaptive' sub-sampling done during training, explained in 2.4.2, was not enough to let the model understand the great variability of this kind of signal. It is also evident that the performances among the different models, apart from the biLSTM, were really homogeneous, therefore, it can be said that this result is mostly explained by the data and not by the specific model architecture. In fact, the nature of such data is strictly connected to time, hence, analyzing the singular peaks on their own (as crops) is probably not enough. More advanced recurrent or transformer based architectures have been showing promising results on very similar tasks [6, 10], presumably meaning more effort in this direction will produce better results.

It is worth noting that the performance expressed by the binary classifier is significantly

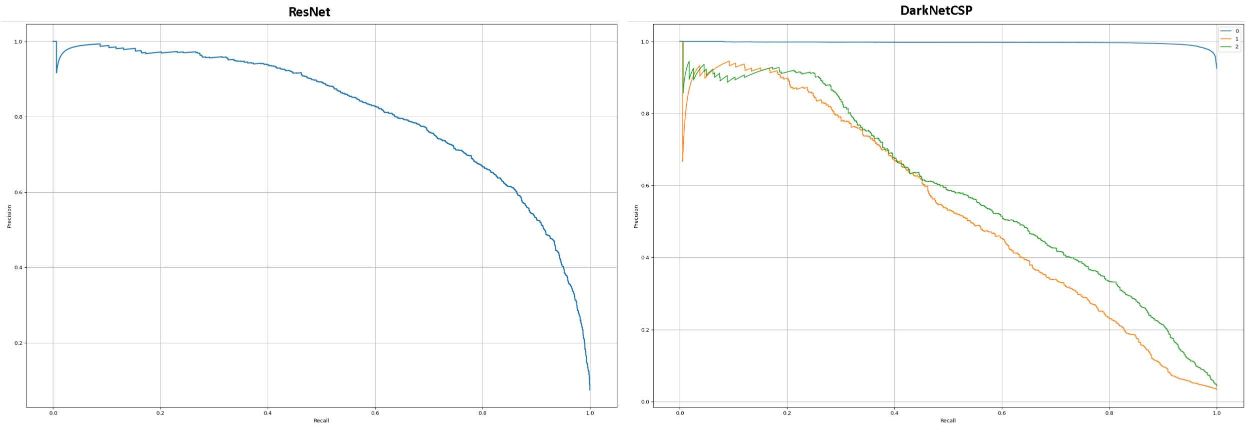

Figure 12: *Comparison between the Recall-Precision curves obtained for the binary (left) and multi-label (right) task. For the binary classifier only the class 1 curve is plotted, while for the multi-target one it is plotted for every class: 0 in blue; 1 in orange and 2 in green.*

better than that shown by the multi-class one. This can be explained by the extreme similarity between the Ventricular and supraventricular contraction. Since the model is trained on classifying each peak by looking only at the crops of the signal containing it, so only considering the peak morphology without considering the time relationship between crops, it is understandable that the model struggles with discriminating V-labelled from S-labelled samples. All the models, within both of the two addressed tasks, were found to be profoundly suffering from overfitting. This is clearly visible in Figure 14, where the progression of the validation loss is plotted against that of the training. It is evident that after almost half of the training period, for both the models, the validation loss begins to increase, while the training one still maintains its descending trend. This is a clear sign of overfitting, as it means that the models are starting to learn the data instead of learning the task. This suggests that further investigation on the model complexity is needed to find the perfect balance between learning and generalization capabilities. By looking at the confidence box-plots (Figure 11), it is clear that the model is much more confident when predicting normal peaks then it is when predicting abnormal ones. This aligns with the obtained metrics which show a general tendency of the model to correctly identify normal peaks, having 93.6% and 93.4% F1-score in binary and multi-class respectively, while having very low precision inside the other classes, 39.9% for class 1 for the binary task, and 34.3% and 38.1% for class 1 and 2 respectively when dealing with the multi-class classification. This result suggests that some thresholding on the confidence score could be done to the model predictions in order to balance the performance of the model. Nevertheless, by looking at the Receiver Operating Characteristic (ROC) and the RecallPrecision curve (RPC), figures 13 and 12 respectively, it is evident that both the models will be very sensible to any alteration of the prediction threshold.

5. Conclusion

Autonomous detection and classification of arrhythmias remains an unsolved challenge, thus an analysis of both a DL and an ML proposal has been carried out to attempt to investigate the potentialities of both worlds. In the end, none of the proposed models really manages to produce any valuable or effective result, meaning more efforts in developing these models is still needed. This work is able to underline the feasibility of applying these tools to such a relevant medical scenario. DL proved to be superior due to its inherit autonomous learning capability that allows hidden, but highly meaningful, relationships within data to be captured. Nevertheless, DL models are still widely accepted within the medical world as they are typically lacking in explainability while ML models tend to be more interpretable due to the hand-crafted nature of the computed features. Luckily, this firewall is beginning to be surpassed and many works are suggesting such an advancement [1, 3]. This work

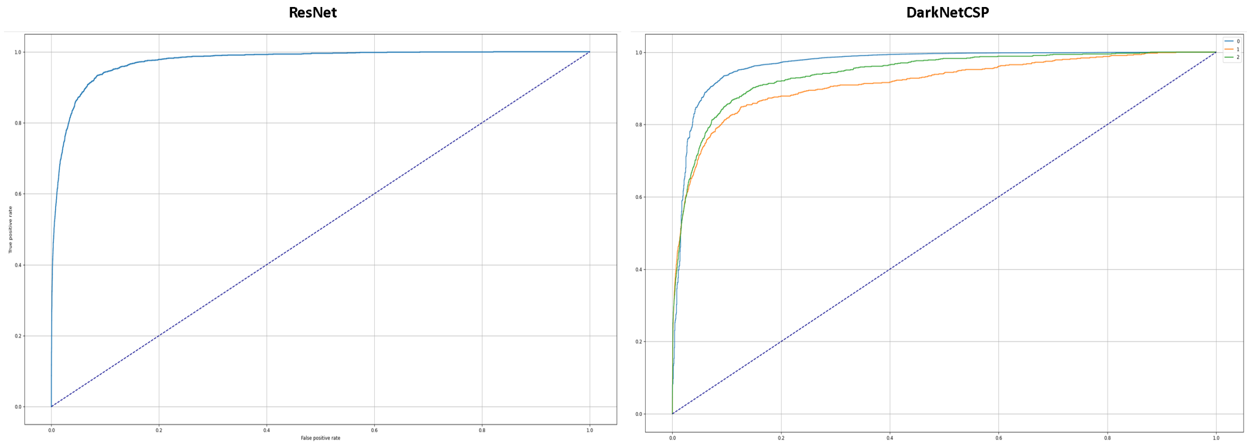

Figure 13: *Comparison between the ROC curves obtained for the binary (left) and multi-label (right) task. For the binary classifier only the class 1 curve is plotted, while for the multi-target one it is plotted for every class: 0 in blue; 1 in orange and 2 in green.*

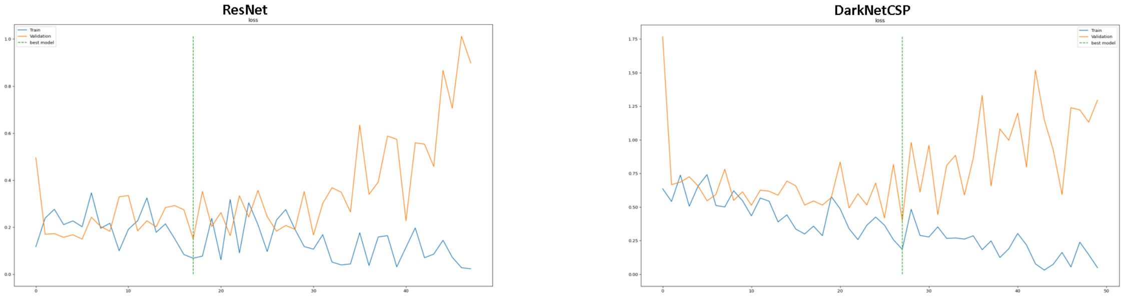

Figure 14: *Loss progression in train (blue) vs. validation (orange) for both ResNet (left) and DarkNetCSP (right). The dashed line represents the best obtained model in term of minimum validation loss. Overfitting can be clearly seen after the dashed line in both the models.*

also highlights the importance of time-related features to correctly classify peaks within a PPG signal; in fact, for the ML paradigm, PTP and A-to-N ratio were found to be the most important (Figure 10), presumably suggesting that, within the DL framework, including the temporal context could certainly be beneficial. In conclusion, we believe this work could act as a valid baseline for future developments. Improvements and more refined parameter optimizations are needed, but the obtained results are promising and show that both the DL an ML frameworks could be used for the task at hand.

6. Appendix

6.1. Other tested DL strategies

To try catching the time relationship between peaks, some transformer based architectures have been investigated. These models have been evaluated only on the validation set and generally show very poor performances. This is mainly due to the unfeasibility of balancing the dataset if not by using a weighted loss, but the so obtained result is just a 'virtual' solution of the imbalance problem and hence insufficient. To exploit such temporal relationship two main strategies were investigated (Table 11). A transformer head was placed after a DarkNetCSP feature extractor to classify crops sequentially inside every patient. Inputs were batched to contain sequential crops into them, fed to the feature extractor to obtain a meaningful embedding of them, fed to the transformer to capture the temporal relationship between them, and finally classified with some dense layers. The model was trained by first freezing the backbone with the weights learned from the crops training loaded and training the transformer, then fine tuning the entire model with very low learning rate of

4e-7.

	Transformer	Unet
Recall	48.2	65.9
Precision	80.2	72.8
F1	60.2	69.2
Accuracy	72.5	78.7

Table 11: *Performances of the 2 investigated strategies on the validation set for the binary task. Apart from accuracy, the metrics are computed on class 1*

The other investigated strategy was inspired by the concept of Region-based CNN. A Unet-like architecture was created to analyze windowed portions of the signals at the same time. The signals were passed to the models not in form of crops, but as fixed-size sequences extracted in a sliding window fashion from every patient. The peaks were then classified by cropping the model output, which was still of the same length as the input, with a small FFN.

6.2. Raw vs. Clean vs. ALL

To better analyse the performance of these models in different scenarios (Table 12), they were trained and tasted also under different conditions and with different kinds of training data. Since the pyPPG [2] package had some functions for extracting the first, second and third derivatives of the signals, the possibilities of using it to enhance the model performance was also considered. Thus, crops from this signals were also extracted and a model was trained to classify peaks given an input containing the crop of the raw signal, that of the filtered one, the first, the second and the third derivatives. Actually, that strategy was only able to balance the prediction towards an higher prediction by lowering the actual recall.

	raw	cleaned	all	all+cleaned
Recall	92.4	95.1	87.4	86.0
Precision	22.2	39.8	29.6	50.1
F1	35.9	56.1	44.2	64.0
Accuracy	83.5	91.7	85.6	89.6

Table 12: *Comparison of the ResNet model over different input data. Apart from accuracy, the metrics refers only to class 1. Raw stands for the dataset containing outliers, cleaned refers to the dataset with the outliers removed and all stands for the usage of all the derivatives signals along with the raw and filtered crop.*

A nice conclusion can be instead be drawn out of this analysis. Indeed,if we look at the results obtained by the model trained on the uncleaned dataset, hence containing outliers, the results are not that far from that obtained by the model trained on the cleaned one. This result suggests that, with further refinements, a robust and effective model could also be develop to handle and classify also portion of the signals corrupted by motion artifacts or any kind of distortions.

7. Bibliography

References

[1] Talal A.A. Abdullah, Mohd Soperi Bin Mohd Zahid, Tong Boon Tang, Waleed Ali, and Maged Nasser. Explainable deep learning model for cardiac arrhythmia classification. In *2022 International Conference on Future Trends in Smart Communities (ICFTSC)*, pages 87–92, 2022.

[2] Márton Áron Goda, Peter Charlton, and Joachim A. Behar. pyPPG: A Python toolbox for comprehensive photoplethysmography signal analysis. *arXiv (Cornell University)*, 9 2023.

[3] Yong-Yeon Jo, Joon myoung Kwon, Ki-Hyun Jeon, Yong-Hyeon Cho, Jae-Hyun Shin, Yoon-Ji Lee, Min-Seung Jung, Jang-Hyeon Ban, Kyung-Hee Kim, Soo Youn Lee, Jinsik Park, and Byung-Hee Oh. Detection and classification of arrhythmia using an explainable deep learning model. *Journal of Electrocardiology*, 67:124–132, 2021.

[4] Zengding Liu, Bin Zhou, Zhiming Jiang, Xi Chen, Ye Li, Min Tang, and Fen Miao. Multiclass arrhythmia detection and classification from photoplethysmography signals using a deep convolutional neural network. *Journal of the American Heart Association*, 11, 4 2022.

[5] Jermana L. Moraes, Matheus X. Rocha, Glauber G. Vasconcelos, José E. Vasconcelos Filho, Victor Hugo C. De Albuquerque, and Auzuir R. Alexandria. Advances in photopletysmography signal analysis for biomedical applications. *Sensors*, 18(6), 2018.

[6] Shu Lih Oh, Eddie Y.K. Ng, Ru San Tan, and U. Rajendra Acharya. Automated diagnosis of arrhythmia using combination of cnn and lstm techniques with variable length heart beats. *Computers in Biology and Medicine*, 102:278–287, 2018.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Andrius Sološenko, Andrius Petrenas, and Vaidotas Marozas. Photoplethysmography-based method for automatic detection of premature ventricular contractions. *IEEE Transactions on Biomedical Circuits and Systems*, 9:662–669, 10 2015.

[9] Simon Stewart, Carole L Hart, David J Hole, and John J.V McMurray. A population-based study of the long-term risks associated with atrial fibrillation: 20-year follow-up of the renfrew/paisley study. *The American Journal of Medicine*, 113(5):359–364, 2002.

[10] Min-Uk Yang, Dae-In Lee, and Seung Park. Automated diagnosis of atrial fibrillation using ecg component-aware transformer. *Computers in Biology and Medicine*, 150:106115, 2022.