A new metric for the classification of cardiac arrhythmias in PPG signals

Maurizio Pandocchi
Politecnico di Milano
Biomedical Engineering
maurizio.pandocchi@mail.polimi.it

Francesca Vacca
Politecnico di Milano
Biomedical Engineering
francesca.vacca@mail.polimi.it

Francesco Santambrogio
Politecnico di Milano
Computer Science and Engineering
francesco2.santambrogio@mail.polimi.it

***Abstract*—This study focused on enhancing cardiac arrhythmia detection through PPG signals, specifically to differentiate normal PPGs from Premature Ventricular and Premature Atrial Contractions, whose high frequency could be an indicator of underlying heart complications or diseases. The research adopted a dual approach, leveraging both Machine Learning and Deep Learning for two types of classification: a binary classification to distinguish normal from both premature contractions, and a ternary classification to distinguish normal, premature ventricular, and premature atrial contractions. Additionally, a novel evaluation metric was developed to offer valuable insights into model performance, namely a weighted f1-score called custom f1. Good performance is generally achieved in binary classification, especially with Ada Boost and Random Forest with 0.92 of custom f1, while ternary classification performed less well, with high values of sensitivity only with premature ventricular contractions, especially with Ada Boost and Random Forest, both 0.92.**

***Index Terms*—Photoplethysmoghram, classification, signal processing, machine learning, deep learning, neural networks**

I. Introduction

Photoplethysmography (PPG) is a technique that assesses the light absorption or reflection within human tissues. This optical signal captures variations in blood volume within the microvascular tissue bed, providing valuable insights into the cardiovascular, respiratory, and nervous systems [1]. PPG holds significance due to its noninvasive nature and cost-effectiveness, making it widely accessible through affordable sensors. Several studies with different applications of PPG signals have been carried out [2], especially focusing on cardiac disorders and arrhythmia such as premature atrial contractions (PACs), and premature ventricular contractions (PVCs). The former are common cardiac dysrhythmia characterized by premature heartbeats originating in the atria, while the latter are early depolarizations of the myocardium originating in the ventricle. Both types of contractions were thought to be relatively benign in the absence of structural heart disease, but recent studies have demonstrated that frequent PACs can be predictors of atrial fibrillation, stroke, and death [3], while frequent PVCs are sometimes paired with left ventricular dysfunction and LV dysfunction progression [4]. Accordingly, PAC and PVC detection and classification in the PPG is essential to prevent such heart diseases. To accomplish such a task, different approaches have been used [5], such as a computerized analysis of the PPG signal, the Poincarè plot method [6], or Machine Learning-based and Deep Learning-based classifiers [5], which are preferred for their scalability and adaptability to different datasets. Some studies have used traditional ML techniques exclusively, like Gil et al. [7], who utilized a Linear Discriminant Analysis (LDA) classifier with a 96.8% accuracy in PVC detection. Others, such as Javed et al. [8], employed neural networks (NNs), specifically Convolutional NNs (CNNs), achieving a 97.9% accuracy in distinguishing between PAC and PVC. Solosenko et al. [9] utilized an Artificial Neural Network (ANN) with backpropagation, achieving sensitivity/specificity rates of 96.05/95.37% and 99.85/99.80% for classifying PVC types. Liu et al. employed Diffusion-CNN (DCNN) to classify six rhythm types, including PAC and PVC, with an overall accuracy of 85%. Additionally, some studies, like that of Yousefi et al. [10], combined NNs with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers, with KNN achieving the highest recognition accuracy of 95% and specificity of 90.4%. This study aims to classify pulses within PPG signals as Normal, PAC, or PVC using both traditional ML and DL classifiers and to compare their performance.

II. Methods

A. Data inspection.

The dataset consists of 105 PPG signals, 62 of which were sampled at 128 Hz, 43 at 250 Hz, each signal lasting about 30 min, and peaks labeled as N, S (PAC), or V (PVC). The task is framed as a twofold classification: binary, aimed at distinguishing normal from abnormal beats, and ternary, to tell the three classes apart. From the labels histogram, it was possible to notice a substantial imbalance in the dataset, with a proportion of 92.83%, 3.93%, and 3.24%. Finally, an initial visual inspection evidenced some corrupted portions of signals causing a distortion both in amplitude and frequency content.

B. Signal pre-processing.

In Fig 1 the general pipeline of signal pre-processing, which will be explained in detail in the following sections, is shown.

- Resampling. A good practice to simplify the processing pipeline [11] was to resample all signals to the same frequency, chosen as the lowest of the two (128 Hz) to preserve the original information.
- Artifact removal. By comparing signals at our disposal and samples from works in literature [12], we concluded that the noise corrupting our signals was not relatable to motion or drift. Therefore, custom techniques were explored: the ones

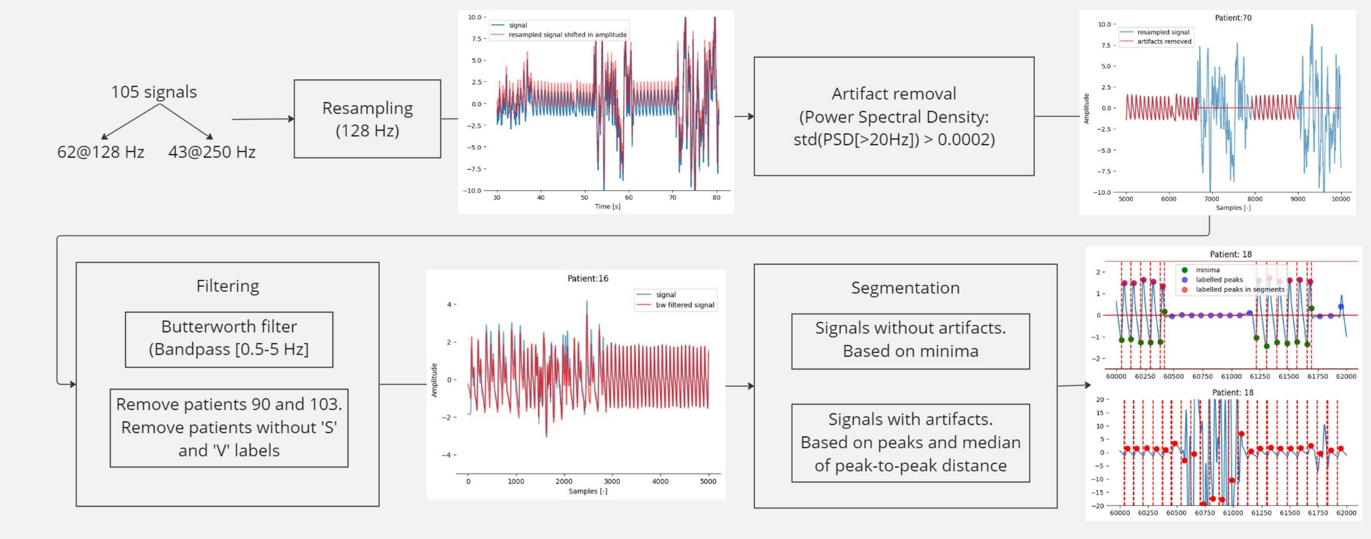

Fig. 1: Proposed pipeline for signal pre-processing

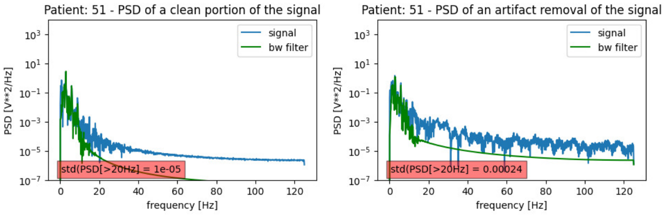

Fig. 2: Periodogram of a clean (left) and noisy (right) portion of signals; the value indicates the standard deviation of the PSD after 20 Hz. A threshold of 0.0002 should indicate a distinction among the two conditions.

providing more repeatable results were filtering according to the Power Spectral Density (PSD) of portions of signals, and anomaly detection through an autoencoder network based on dense layers. After some trial and error, it was possible to establish that the portions having a standard deviation of PSD higher than a threshold of 0.0002 after 20 Hz of frequency, corresponded to noise; this allowed to preserve the frequency content of PPG pulses which is surely lower than 20Hz, while the noise comprehends a wider range. From Fig 2 it is possible to appreciate the periodogram's behaviour.
For linearity of the dissertation, the autoencoder will be reported after the segmentation section.

- Filtering. To reduce the noise content that was not captured with artifact removal while preserving the clinical bandwidth of PPG signals [13], [14], we applied a band-pass Butterworth filter of the third order with [0.5-5 Hz] bandwidth. Additionally, 2 signals were removed because of their visibly deteriorated quality, and 12 signals were excluded since they consisted of only N-labelled peaks.
- Segmentation. The preprocessing could finally terminate with signal segmentation, namely signal chunks, each identifying a single pulse, whose frequency reflects as much as possible the heart rate of the patient. Two segmentation techniques could include the assumption of the presence or absence of artifacts in the signal. Whereas the absence of artifacts, as in the pipeline described so far, guarantees an alternation between peaks and minima, the presence of artifacts may cause anomalous behaviors. Therefore, signals without artifacts were segmented through a semi-automatic method that was carefully tailored to particular cases. Signals with artifacts, instead, were segmented based on the median of peak-to-peak distances of each patient, which serves as an estimation of their heart rate less sensitive to outliers, so that the onset position is imputed at one-third of the median to the left of the peak, while the offset position is imputed at two-thirds of the median to the right of the peak. The choice of the onset and offset positions was due to the asymmetry of the pulses with respect to their peaks. Despite being less analytical than the previous, this method has proved to identify almost the same segments as the other method. The segmentation was applied to create blocks of three consecutive segments as well while we were exploring the best alternative to provide to deep learning models. Filtering was applied, to eliminate segments with: amplitude higher than 2.5 or lower than -2.5, which are approximately the amplitude limits of physiological pulses ii) durations out of physiologically plausible ranges (30-220 BPM), iii) labeled peaks that are different in time and amplitude, more than a certain threshold, from the actual maximum, iv) number of actual maxima higher than 1 (excluding cases of segments with the dicrotic notch, that ideally show an absolute maximum followed by a relative maximum).
The consecution of pre-processing steps brings to a reduction of the number of segments by [62.71%, 68.05%, 58.38%] for the three types of pulses respectively.

C. Anomaly detection.

As mentioned above, an autoencoder was used as an anomaly detector to select artifact segments. The best perform-

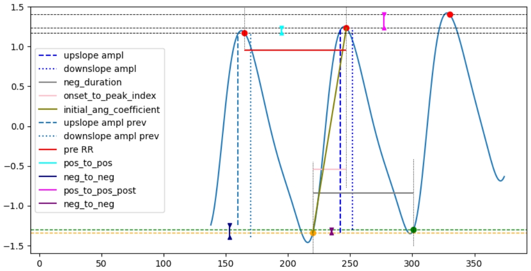

Fig. 3: Salient peak-to-peak features; from left to right: previous, current and following segment.

ing was a FFNN autoencoder, which was trained using only clean segments obtained after the first segmentation method, and then tested on noisy data (the segments filtered out after the pre-processing steps) to find a threshold. This provided a separation between clean and noisy segments when applied to a complete signal segmented with the second method.

D. Feature extraction.

Our efforts were devolved to extract features that could comprehensively describe and possibly distinguish classes, by combining inspirations from literature and some custom features.

- Statistics-based features: to skewness and kurtosis [15], we added mean, median, variance, and standard deviation of segment values.
- Peak features: area, maximum and minimum of the first derivative, maximum and minimum of the second derivative [2].
- peak-to-peak features: those combining values from the previous, the current, and the following segment for each segment, [13], [16], [14]. Fig 3 summarizes the salient peak-to-peak features; features based on distances (in amplitude or indexes) were normalized in a subject-specific manner, as suggested in [11].

E. Feature pre-processing.

The processing steps here described were defined on the training set and then applied to test and validation sets. Firstly, we proceeded with the removal of outliers, detected as those values having a z-score higher than 3 after applying standard scaling, and successively re-assigned to NaN. The substitution of NaN values with the median of the relative feature resulted in a drastic reduction of variability, therefore the imputation through KNN Imputer was preferred. Once the outliers were removed, various normalization methods were tested, including those biased by the presence of values very distant from the distribution such as the min-max scaling, since we had previously re-assigned the outliers. Z-score normalization has the effect of standardizing the mean and variance of features, therefore any feature selection according to a threshold on the variance would be unfruitful. On the other side, features should be somehow brought on the same scale to be truly comparable, therefore a min-max normalization was applied before the selection.

F. Feature selection.

Features dimensionality space was reduced according to two filtering criteria: variance lower than 0.01 and correlation with another feature higher than 0.7 (among the two, we dropped the one with the highest mean correlation with the others). Wrapper methods were discarded because too computationally demanding, while PCA provided less convenient results. The feature selection step led to a reduction of the features space from 40 to 12.

G. Data preparation for models.

To address the stratification of composite labels, an Integer Linear Programming (ILP) model was exploited, ensuring each subjects's segments were consistently assigned to a single set while minimizing error from an ideal 90/10 split. While carrying out our work we didn't fix the seed of the split, in order to evaluate everytime a new combination and reach more stability in the models performance. Furthermore, to mitigate dataset imbalance, undersampling of N-labeled signals and features was conducted. In contrast, for the under-represented S- and V-labeled sets, upsampling techniques such as random bootstrapping and SMOTE were explored, with SMOTE's synthetic data generation proving the most effective.

H. Custom metric and confidence.

Since accuracy could not be considered as a relevant metric due to the unbalance of the classes, *Precision* and *Sensitivity* were reported in tables, together with the custom metric. Such new metric took into account the proportions between the classes and was defined as a weighted average of the f1 scores of each class relative to the other two.

The weights used to compute the new metric were defined to give more importance to the underrepresented classes:

$$weights = \left[\left(1 - \frac{\#N}{set\ size}\right), \left(1 - \frac{\#V}{set\ size}\right), \left(1 - \frac{\#S}{set\ size}\right) \right] \quad (1)$$

where #N, #V and #S are the quantity of respectively N-, V- and S-labelled components in set, either training, validation, or test, and set size is the corresponding set size. This vector is then normalised.

At this point, having defined the vector $f1-score$ as the vector of the $f1-score$ of each class relative to the others, the metric is calculated as the scalar product between this vector and the vector of weights:

$$f1 - score = \left[f1_{N\ vs\ (S,V)}, f1_{V\ vs\ (N,S)}, f1_{S\ vs\ (N,V)} \right] \quad (2)$$

$$custom\ f1 - score = weights^T \cdot f1 - score \quad (3)$$

This specific formulation considers three classes, but the metric can be applied to binary classification as well.

The new metric was used both during the training phase and to evaluate the model. Two dedicated callbacks, Early Stopping and the Reduction of the Learning Rate on Plateau, were redefined on the new metric to prevent overfitting.

Once the prediction was made on the test set, it was necessary to assess the confidence of these predictions for the

V and S classes. For ML, it was defined as the probability of the prediction, whereas for DL it was considered the difference of the output softmax probabilities of the two most probable predicted classes.

I. Machine learning.

To address the classification tasks, traditional ML classifiers were exploited at first. On the one hand, simpler and less computationally demanding models: SVM, LDA used also by Gil et al. [7], and KNN; on the other, ensemble models: Random Forest, Ada Boost, and Gradient Boosting. To obtain the best predictive models, hyperparameters tuning via tenfold cross-validation with the train_val set was performed. Among the different parameters, those achieving the highest custom f1-score were selected. As regards the performance, the most important metrics were then considered for each classifier, with a special focus on precision, recall, and weighted f1 score due to the dataset imbalance, as well as the ROC curve with the related AUC.

J. Deep learning.

Initially, a VGG network and a ResNet were used, each allowing a long depth to be reached. Subsequently, an approach inclusive of both DL and hand-crafted features was applied, with Wide ResNet and Wide DenseNet networks. The features previously used for traditional ML classification, in the case of the Wide ResNet, were processed through a dense layer and then concatenated to the output of the convolution, before classification. In the Wide DenseNet, instead, the dense layer was removed.

A further test was performed with the addition of Convolutional Block Attention Modules (or CBAM), referred to as Wide ResNet CBAM. CBAM blocks are intermediate attention feature maps used for adaptive feature refinement [17].

Finally, a network using LSTM, multi-head attention, and convolutional layers was utilized. Multi-head attention is a mechanism allowing the model to concentrate on different information from the input segment at different positions [18].

All these networks, initially conceived specifically for the three-class classification, were used for binary classification as well, just by modifying the number of output neurons from 3 to 2.

III. Results

A. Machine learning.

a) Binary classification: The outcome of the binary classification performance with traditional ML classifiers is displayed in TableI.

TABLE I: Precision and Sensitivity of the binary classes, and $custom\ f1 - score$

Model	Pr N	Pr S&V	Se N	Se S&V	**custom f1**
SVM	1.00	0.76	0.98	0.95	**0.85**
LDA	0.99	0.88	0.99	0.93	**0.91**
KNN	1.00	0.82	0.98	0.94	**0.89**
Random Forest	0.99	0.88	0.99	0.93	**0.92**
Ada Boost	0.99	0.89	0.99	0.93	**0.92**
Gradient Boost	0.99	0.81	0.98	0.93	**0.90**

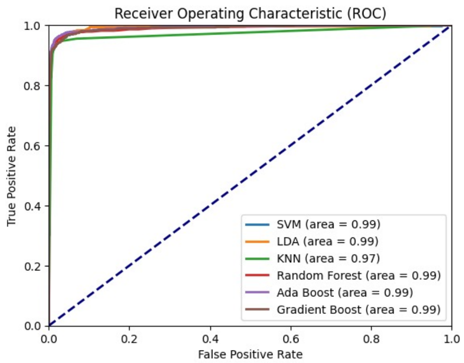

Fig. 4: ROC curves for all ML classifiers computed on the internal test set.

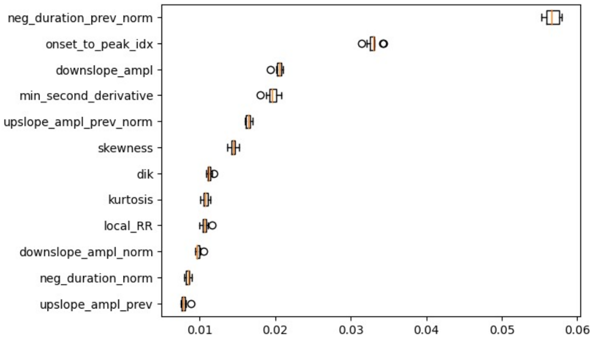

Fig. 5: Features permutation importance for binary classification with AdaBoost

As shown in the table, almost all the models exceeded 0.8 in every metric, except for the Precision for the premature pulses of the SVM (0.76). The highest values of custom f1-score were achieved by the ensemble models, especially Random Forest and Ada Boost, both with 0.92, while the performance of the other models was slightly lower. The achieved mean confidence exceeded in all models 0.9 value for both labels. In particular, the highest confidence values were achieved by LDA with 0.98 and 0.93 for normal and premature contractions classes respectively, while the lowest ones were achieved by Ada Boost with 0.95 and 0.89 respectively.

Below we report the ROC curves on the internal test set for all ML classifiers (Fig 4), and the representation of the features permutation importance for the binary classification (Fig 5).

b) Ternary classification: The outcome of the ternary classification performance with traditional ML classifiers is displayed in TableII.

TABLE II: Precision and Sensitivity of the S and V classes, and $custom\ f1 - score$ for each used model

Model	Pr S	Pr V	Se S	Se V	**custom f1**
SVM	0.46	0.58	0.40	0.92	**0.47**
LDA	0.12	0.30	0.11	0.32	**0.17**
KNN	0.57	0.62	0.47	0.81	**0.54**
Random Forest	0.79	0.70	0.53	0.92	**0.65**
Ada Boost	0.77	0.71	0.54	0.92	**0.65**
Gradient Boost	0.64	0.63	0.52	0.86	**0.60**

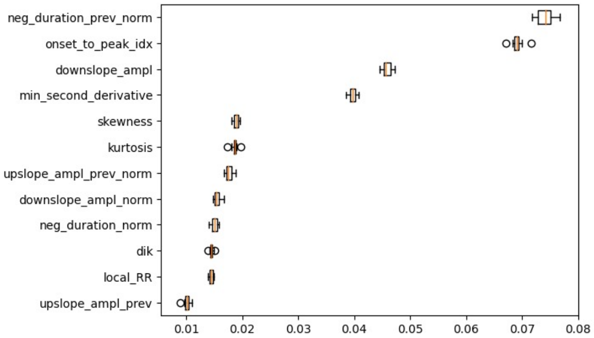

Fig. 6: Features permutation importance for ternary classification with AdaBoost

As shown in the table, all the models did not exceed 0.65 in the custom f1-score. Only the sensitivity for the PVC was high in almost every model, except for the LDA which performed poorly. The highest values of custom f1-score were achieved again by the ensemble models, especially Random Forest and Ada Boost both with 0.65, while the performance of the other models was much lower, with only KNN exceeding 0.5 of custom f1-score. As regards confidence, all the models had a high score for normal pulses, around 0.95 or above, though very low values for PAC, all between 0.4 and 0.5. Differently, the mean probability of the PVC pulses belonging to the proper class was 0.8 in the SVM, around 0.66 for the Random Forest and Ada Boost, while only 0.5 for the LDA classifier.

Below we report the representation of the features permutation importance for the ternary classification (Fig 6).

B. Deep learning.

a) Binary classification: The binary classification of Normal segments against Ventricular and Atrial segments was successful, and reached a value of the $custom\ f1-score$ of 0.80, with a recall for the two classes equal to 0.96 and 0.97 respectively. The model of choice was the Wide DenseNet, in which the features of the inputs were concatenated to the output of the last convolutional layer.

The mean confidence achieved for the two classes is 0.94 for the N, and 0.80 for S and V.

TABLE III: Precision and Sensitivity of the binary classes, and $custom\ f1-score$

Model	Pr N	Pr S&V	Se N	Se S&V	**custom f1**
Wide DenseNet	1.00	0.67	0.96	0.97	**0.80**

b) Ternary classification: At first, only segment inputs were considered, but given the poor results of 47% and 52% for ResNet and VGG respectively, hand-crafted features were concatenated at the end of the convolutional network. However, the results of the implemented Wide ResNet and Wide DenseNet didn't show a substantial improvement in classification, reaching values of 49% and 45% respectively.

To keep track of particular portions of the input signal, a Wide ResNet CBAM architecture was used, demonstrating to have better performances than the Wide DenseNet, but not than the Wide ResNet.

A further approach involved treating each segment as a single-featured time-series, by using an LSTM network together with convolution. Again, however, the network had difficulty in distinguishing between S and V, despite the use of Multi-Head attention.

Overall, the network with the best performance was the VGG, which also had the greater distinction rate between S and V classes. Anyway, the level of confidence was rather low for both of them, 0.38 and 0.46 respectively. Class N segments were instead classified as such with a confidence of almost 0.92.

TABLE IV: Precision and Sensitivity of the S and V classes, and $custom\ f1-score$ for each used model

Model	Pr S	Pr V	Se S	Se V	**custom f1**
VGG	0.53	0.37	0.55	0.62	**0.52**
LSTM	0.37	0.40	0.54	0.44	**0.45**
ResNet	0.39	0.35	0.62	0.50	**0.47**
Wide ResNet	0.42	0.39	0.60	0.56	**0.50**
CBAM	0.38	0.43	0.60	0.50	**0.48**
Wide DenseNet	0.37	0.35	0.53	0.51	**0.45**

IV. DISCUSSION

Concerning the binary classification task, both ML and DL classifiers performed well, highlighting a high discriminant power between the features of normal pulses and those of premature contractions. The best performance was achieved by the traditional ML Ada Boost classifier with 0.916 of the $custom\ f1-score$ value, almost leveling off the performance of Yousef et al. [10]. This reflected the expectations since ensemble models are expected to have better predictive performance in complex and unbalanced datasets with non-linear relationships. As regards the ternary classification, the overall performance gets worse, underlying the fact that PAC and PVC pulses have less discriminant features. Moreover, the high imbalance of the dataset favoring normal peaks, together with the highly noisy signals, did constitute an obstacle to extracting enough information to distinguish between PAC and PVC signals and achieve state-of-the-art performance [7]–[9].

As regards the classification with DL models, the overall worse performance with respect to Machine Learning is due to the inability of the networks to distinguish precisely the two classes S and V, which are confused with each other. This is probably due to the fact that the morphology of the two signals appears to be very similar, making it difficult to distinguish them based on the segments alone. The addition of the features didn't lead to any substantial improvement, probably due to the bias introduced by the information extracted from the segments via convolution. In addition, all the networks had as input a tensor of segments that needed to be homogeneous in length, meaning that all the segments had to be resampled to the same number of samples, which introduced an under-representation of longer segments. The resampling length chosen was in fact equal to 66, equal to the length of the shorter segment. This may have introduced a further error in the classification or shrank too much the sequences for the networks to learn enough from them.

V. Conclusion

This study focused on detecting cardiac arrhythmias, specifically premature atrial contractions (PAC) and premature ventricular contractions (PVC), in photoplethysmogram (PPG) signals. The classification task is approached both as binary (PAC/PVC vs. non-PAC/PVC) and ternary (PAC, PVC, and non-PAC/PVC) classification of individual PPG pulses.

Our methodology involved integrating information from the PPG signals themselves along with hand-crafted features, which were subjected to dimensionality reduction and normalization to facilitate subsequent analysis.

The challenge arose from the need to extract features from segments of the PPG signals, which being very short-lasting may not provide sufficient information for deep learning models to effectively discriminate between classes. As a result, traditional machine learning models outperformed deep learning models in both binary and ternary classifications. Model performance was evaluated using a custom metric tailored to the dataset's characteristics, along with a custom confidence level. Our best-performing model, AdaBoost, achieved custom metric values of 0.92 for binary classification and 0.65 for ternary classification.

Future directions for this study could focus on improving the integration of information from PPG signals and hand-crafted features to enhance classification precision and recall.

References

[1] John Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1, 2007.

[2] M. A. Almarshad, M. S. Islam, S. Al-Ahmadi, and A. S. BaHammam. Diagnostic features and potential applications of ppg signals in healthcare: a systematic review. *Healthcare*, 2022.

[3] David Conen, Martin Adam, Frederic Roche, Jean-Claude Barthelemy, Denise Felber Dietrich, Medea Imboden, Nino Künzli, Arnold von Eckardstein, Stephan Regenass, Thorsten Hornemann, et al. Premature atrial contractions in the general population: frequency and risk factors. *Circulation*, 126(19):2302–2308, 2012.

[4] Paul L Eugenio. Frequent premature ventricular contractions. *Cardiology in review*, 23(4):168–172, 2015.

[5] Neha, HK Sardana, R Kanwade, and S Tewary. Arrhythmia detection and classification using ecg and ppg techniques: A review. *Physical and Engineering Sciences in Medicine*, pages 1–22, 2021.

[6] Dong Han, Syed Khairul Bashar, Fahimeh Mohagheghian, Eric Ding, Cody Whitcomb, David D McManus, and Ki H Chon. Premature atrial and ventricular contraction detection using photoplethysmographic data from a smartwatch. *Sensors*, 20(19):5683, 2020.

[7] Eduardo Gil, Pablo Laguna, Juan Pablo Martínez, Oscar Barquero-Perez, Arcadi Garcia-Alberola, and Leif Sörnmo. Heart rate turbulence analysis based on photoplethysmography. *IEEE transactions on biomedical engineering*, 60(11):3149–3155, 2013.

[8] Aaisha Javed, Muhammad Usman Akram, and Norah Saleh Alghamdi. Signal processing and deep learning based smartwatch photoplethysmography data classification of atrial fibrillation, premature atrial and ventricular contraction. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 18–23. IEEE, 2023.

[9] Andrius Solosenko and Vaidotas Marozas. Automatic premature ventricular contraction detection in photoplethysmographic signals. In *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, pages 49–52. IEEE, 2014.

[10] Mohammad Reza Yousefi, Mahdi Khezri, Razieh Bagheri, and Reza Jafari. Automatic detection of premature ventricular contraction based on photoplethysmography using chaotic features and high order statistics. In *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–5. IEEE, 2018.

[11] A. Solosenko, A. Petrenas, and V. Marozas. Photoplethysmography-based method for automatic detection of premature ventricular contractions. *IEEE*, 2015.

[12] D. Pollreisz and N. TaheriNejad. Detection and removal of motion artifacts in ppg signals. *Springer*, 2019.

[13] S. Dhar, A. Chakraborty, D. Sadhukan, S. Pal, and M. Mitra. Effortless detection of premature ventricular contraction using computerized analysis of photoplestysmography signal. 2022.

[14] E. Sabeti, N. Reamaroon, M. Mathis, J. Gryak, M. Sjodin, and K. Najarian. Signal quality measure for pulsatile physiological signals using morphological features: Applications in reliability measure for pulse oximetry. *Elsevier*, 2019.

[15] F. Mohagheghian, D. Han, A. Peitzsch, N. Nishita, E. Ding, E. L. Dickson, D. DiMezza, E. M. Otabil, K. Noorishirazi, J. Scott, D. Lessard, Z. Wang, C. Whitcomb, K. V. Tran, T. P. Fitzgibbons, D. D. McManus, and K. H. Chon. Optimized signal quality assessment for photoplethysmogram signals using feature selection. *IEEE*, 2022.

[16] E. Gil, P. Laguna, J. P. Martinez, O. B. Perez, A. G. Alberol, and L. Sornmo. Heart rate turbulence analysis based on photoplethysmography. *IEEE*, 2013.

[17] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.