Classification of PACs and PVCs in PPG using Convolutional Neural Networks

Leonardo Cesani, Martina Fervari, Andrea Zanette
Report for the Applied AI in Biomedicine Project, Politecnico di Milano
Email: {leonardo.cesani, martina.fervari, andrea.zanette}@mail.polimi.it

***Abstract*—This report presents a comprehensive approach towards the development of a classification model for premature atrial contractions (PACs) and premature ventricular contractions (PVCs) in PPG signals, utilizing neural networks. The study proposes an advanced machine learning technique to enhance the accuracy of distinguishing between anomalies in cardiac rhythms originating in the atria and ventricles, respectively. Through the exploitation of convolutional neural networks along with the integration of problem-specific features, the model aims to achieve good performances in classifying PACs and PVCs against normal sinus rhythm.**

***Index Terms*—PPG, PAC, PVC, Classification, Neural Networks, CNN**

I. Introduction

Photoplethysmography (PPG) is an optical technique used to detect volumetric changes in blood in peripheral circulation. It is a low-cost and non-invasive method that makes measurements of the pulse rate at the surface of the skin. This report aims to explore the development of a model to address a classification problem for PPG signals. Specifically, the task involves assigning a label to each systolic peak position of a PPG signal, indicating whether the patient presents a normal cardiovascular behavior or a pathology, such as *premature ventricular contraction* (PVC) or *premature atrial contraction* (PAC).

A. Photoplethysmography

A *photoplethysmogram* (PPG) is an optically obtained plethysmogram used to detect blood volume changes in peripheral tissue microvasculature. Obtained with a pulse oximeter, it measures changes in light absorption as the heart's pressure pulse alters skin volume. PPG waveforms vary between subjects and depend on the attachment method and location of the pulse oximeter.

B. Abnormal Heart Beats

Premature ventricular contractions (PVCs) are abnormal heartbeats originating in the ventricles, disrupting the regular heart rhythm. On the other hand, premature atrial contractions (PACs) are similar premature beats, but they occur in the atria. While PVCs and PACs generally don't cause heart damage and may occur in healthy individuals without known heart disease, frequent PVCs can be linked to left ventricular dysfunction, and frequent PACs are associated with an increased risk of stroke.

II. Materials and artifacts

The dataset consists of a set of 105 PPG signals, each obtained from a different patient and sampled at different frequencies: 62 signals were sampled at f_s = 128 Hz, while 43 were sampled at f_s = 250 Hz.

A. Dataset

The dataset stored in three different files for each patient:

- PPG files: Contain the data about the PPGs. The information of the sampling frequency and the patient's identification number are included in the file's name;
- Annotations: Each file in this category contains annotations for every heartbeat within the signal. Normal beats are denoted as 'N', ventricular beats as 'V', and sopraventricular beats as 'S';
- Systolic peak position: These files contain annotations for each peak within the signal. Each peak is associated with a label found in the annotations file;

In Table I the dataset proportions are presented. It is evident that the dataset exhibits a substantial skewness, predominantly in favor of the "normal" class, which represents the majority of annotated beats. Additionally, it is possible to observe that the abnormal beats in signals sampled at 250 Hz are very underrepresented, comprising only 0.51% of the total beats. This imbalance in class distribution will be duly considered in the construction of the classification model.

TABLE I
Proportions of the datasets.

	Sampled @128 Hz	**Sampled @250 Hz**
Number of signals	62	43
Signal Lengths (minutes)	30	30
Normal Beats	131828 (53.46%)	97101 (39.37%)
Abnormal Beats	16433 (6.66%)	1252 (0.51%)
Ventricular Beats	7605 (3.08%)	389 (0.16%)
Sopraventricular Beats	8828 (3.58%)	863 (0.35%)

B. Outliers

After exploring the dataset, certain outliers were identified and recognized as artifacts. These artifacts are undesired distortions present in the PPG waveform, not accurately representing physiological changes in blood volume. In PPG signals, artifacts are commonly introduced due to body movements or sensor misplacement, leading to fluctuations in the signal unrelated

to changes in blood volume. The detection of artifacts was facilitated during data plotting, as depicted in Figure 1, which also enables the visualization of ventricular and supraventricular heart beats.

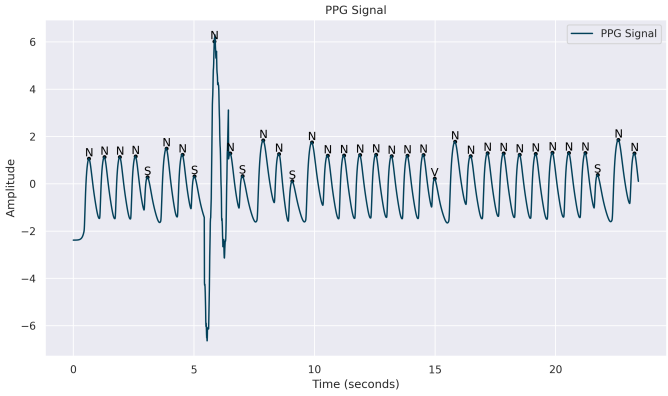

Fig. 1. Ventricular beats, sopraventricular beats and artifacts

The presence of outliers in the dataset may strongly influence the performance of the model, hence it is necessary to carefully detect and remove them.

III. Pre-Processing

In order to feed the model with proper data, pre-processing is needed. A scheme providing the pipeline that has been applied can be observed in Figure 2.

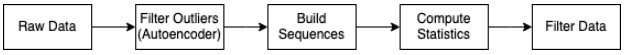

Fig. 2. Pre-processing Pipeline

A. Autoencoder

The initial approach to automatically detect outliers involved the implementation of a *denoising autoencoder* proposed by Vincent et al. in 2008 [6]. This model is designed as an autoencoder trained to reconstruct noisy signals. Given the unavailability of cleaned signals, we opted to generate synthetic PPG examples using *NeuroKit2* [5]. To train the denoising autoencoder effectively, the following steps were applied:

1) Construct the dataset using NeuroKit2 to closely resemble real data;
2) Construct sequences for input to the model;
3) Create a noisy version for each sequence;
4) Feed the noisy sequences to the model, training it to reconstruct the original signal. This involves using the noisy signal as input and expecting the clean signal as output;

After finding the optimal parameters for the model, it has been used to filter outliers in the dataset. Due to the fact that the denoising model has been trained on synthetic data, it was not capable of reconstructing the artifacts of the dataset. Considering these issues, it has been necessary to change paradigm.

A new autoencoder has been trained in order to perfectly reconstruct the input signal, without the noisy counterpart. Again, the model has been properly tuned and then applied to the dataset. In order to detect outliers, a new method has been employed:

1) Divide each signal into windows with a proper dimension to be fed to the autoencoder;
2) Feed the autoencoder with each window and produce the output;
3) Consider the mean squared error between the input and the output of the autoencoder. If the MSE is high, the autoencoder is not capable of reconstructing the signal; hence, an artifact has been found;

Figure 3 shows the effect of applying the autoencoder model to the dataset.

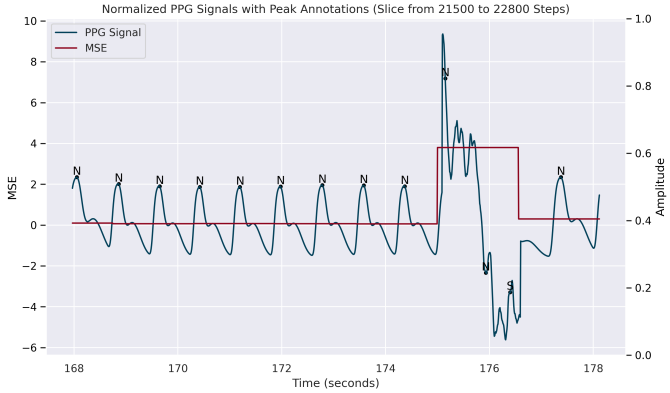

Fig. 3. The application of the autoencoder on the PPG signal

After applying a threshold to the MSE values, all samples exceeding the threshold are set to zero. This strategy proves beneficial in the subsequent features extraction phase, where samples characterized by extended sequences of zeros will be filtered out.

B. Beat Segmentation and Cleaning

A beat-by-beat segmentation was applied to the filtered signals in order to create a dataset containing only the single beats. For each patient, the whole signal was segmented based on the positions of the different systolic peaks, creating a window of fixed length around it. In particular, the window was translated of around one third to the right in order to set the beginning of the beat on the pulse onset and to include together both the systolic and the diastolic peaks. This is an essential step for ensuring the confidence and fidelity of the classification model. After the beat segmentation step, to minimize the effect of high-frequency noise, the dataset was de-noised using a *Butterworth* band-pass filter having a cut-off frequency ranging from 0.1 to 7 Hz. The use of a bandpass filter not only facilitates effective removal of the signal noise elements but also helps to maintain the clinical bandwidth of the PPG signal.

Moreover, normalization was applied with *MinMaxScaler* for each single beat in order to ensure comparability between different signals and to make the model more accurate and

faster to train, ensuring the chances of getting stuck in local optima. To maintain consistency and high accuracy, each beat was checked whether the number of consecutive zeros, caused by the thresholding on the MSE (Mean Squared Error) of the autoencoder, was smaller than 20, otherwise the corresponding window was thrown away from the analysis.

Finally, the training dataset was created by the filtered and normalized beats and their corresponding first derivative.

C. Features Extraction

Since the presence of a pathology causes irregularity in the heartbeat, some statistical features regarding the morphology of the PPG waves were selected in order to better capture the behavior of the different types of signals.

In particular, as proposed by S. Dhar et al. in 2022 [3], the features of kurtosis and skewness has been chosen to describe the morphological changes in the PPG signal due to PVC and PAC, where kurtosis index measures the deviation from a Gaussian distribution, whereas skewness is an indicator of the symmetry (or asymmetry) of the data values around the sample mean. By following the method proposed by S. Dhar et al. in 2022 [3], the initial idea was to implement a methodology that, based on a simple threshold, was able to distinct the normal beat from the pathologic ones just by looking at the differences of kurtosis and skewness among the beats.

Afterwards, a second model would have been implemented to classify each signal, identified as abnormal by the previous model, as affected by PAC or PVC. However, this procedure could not be developed using the available dataset since a fluctuation in the metrics can be appreciated by looking to a single beat compared to the others of a same patient, and not by looking at the entire dataset of beat windows, since each individual has a specific cardiological morphology of the heartbeat. For this reason, it was preferable to consider the differences of the different metrics among the peaks of the same patient. In particular, the chosen features were:

- differences of kurtosis of a peak window with respect to the previous and to the next peak, called DIK_PRE and DIK_POST;
- differences of skewness of a peak window with respect to the previous and to the next peak, called DIS_PRE and DIS_POST;
- distance of a systolic peak position from the previous one and from the next one, called DIP_PRE and DIP_POST;
- differences of amplitude of a peak window with respect to the previous and to the next peak, called DIA_PRE and DIA_POST;

The following step was to check the presence of remaining outliers, not previously removed by the autoencoder, which might present values of the considered metrics significantly out of range with respect to the rest of observations. Specifically, the following thresholds have been used to clean up and harmonized the training dataset:

- regarding kurtosis, the threshold is 25 on DIK_POST;
- regarding peak distances, the threshold is 250 both on DIP_PRE and DIP_POST, as shown in Figure 4;
- regarding amplitude, the threshold is 55 both on DIA_PRE and DIA_POST;

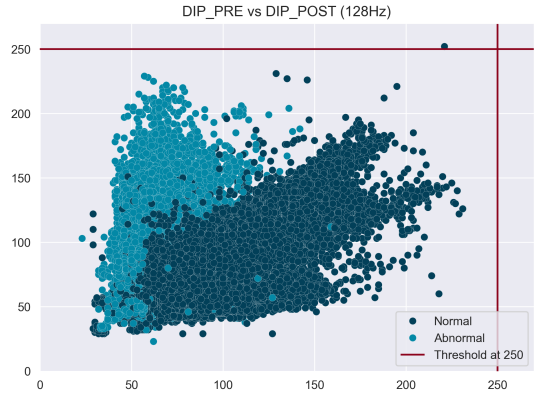

Fig. 4. The threshold for the DIP_PRE and DIP_POST features

A crucial aspect of the pre-processing step was to select a proper approach to treat the signals characterized by different sampling frequencies, in order to enhance the quality of the dataset. In this framework, the 250 Hz signals were resampled to convert the frequency to 128 Hz, as the rest of the beats, and all the features were scaled to allow for signal comparability. This meticulous approach ensured the generation of consistent and meaningful data which can be used as a reliable training set for the neural network model.

Finally, considering the substantial predominance of the N class and the higher likelihood that certain outliers may have persisted through the pre-processing phase, an *Isolation Forest* technique was introduced at the end of the pipeline to further refine the signal dataset. This method strategically isolates anomalies by constructing a tree-based structure, identifying instances that deviate significantly from the norm. It serves as an additional layer of data refinement, aiming to enhance the overall quality of the dataset by effectively filtering out outliers that could potentially impact the accuracy of the subsequent heartbeat classification model.

In the Appendix A it is possible to find a block scheme representing the complete pipeline that has been used to process data.

IV. The Model

The developed model is designed to emulate a ResNet architecture (He, et al., 2015 [4]) with specific modifications tailored for one-dimensional convolutions. Additionally, the model has been expanded to incorporate attention blocks and integrate computed data features. These features, derived from the input data, play a crucial role in enhancing the model's overall performance. In Figure 5 a block-scheme of the model is presented.

A. The ResNet

The usage of *ResNet* offers several advantages. By introducing residual connections, ResNet mitigates the vanishing gradient problem, ensuring more effective training of deep networks.

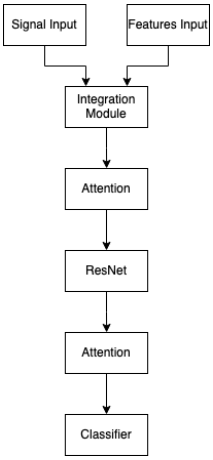

Fig. 5. The model

The modular nature of ResNet's architecture further facilitates scalability and adaptability, being crucial in the model selection phase, allowing for the optimal choice of ResNet building blocks and, consequently, the fine-tuning of the overall network's parameters.

B. The Attention Block

To improve the architecture with an attention mechanism, it has been decided to implement the model a *CBAM* module, proposed by Woo et al., 2018 [7]. In this module, the Channel Attention prioritizes channel importance through global max and average pooling, enhancing feature representation. Simultaneously, Spatial Attention identifies and emphasizes informative spatial regions, contributing to improved localization. The combined impact of both attentions enables the network to adapt dynamically to input characteristics, enhancing model performance.

C. Features Integration

The features detailed in Sec. III-C are incorporated into the model through the addition of a second input. This input is then joined with the signal input to merge both information branches into a unified block. Achieved through concatenation and adjustment of tensor sizes, this approach, as opposed to ensemble models or merging at the classification level, enables the classifier to focus on how different types of information from the signal and features contribute to enhancing classification quality.

D. Model implementation

The ResNet model, along with its constituent blocks, has been implemented using the *Keras API* within a Python class. This implementation provides a convenient and efficient solution, allowing for the rapid instantiate of multiple models without unnecessary repetitions. The Python class includes a constructor that accepts various parameters, such as input shape, the number of classes, and other hyper-parameters like the quantity of CBAM blocks. Prior to the use of the training and evaluation methods, the implemented class provides the option to preprocess input data along with corresponding labels. Additionally, if requested, this method can internally handle the data splitting process into training and testing sets. A detailed exploration of various model variants has been conducted to identify optimal settings for the specific task at hand. Since each one has its own peculiarities, some sub-classes have been designed to override specific methods, ensuring compatibility with the same interface as the base model.

E. Normal vs Abnormal model

An initial use of the model had been for the binary classification task between normal and abnormal beats. All the 8 features, the signal and its derivative were given as input.

A variety of loss functions were experimented to identify the most effective one. After thorough testing, the *Kullback-Leibler* Divergence loss function was ultimately selected. It measures the divergence between predicted and actual probability distributions so the model is optimized to minimize the difference between these two.

To find the optimal setting of the model, Optuna [1] was employed in all the experiments. Throughout an iterative process, this framework systematically explores the hyper parameters space to identify the configuration that maximizes the model's performance.

After determining the optimal configuration for the model, a robust evaluation process was conducted through a cross-validation test with 10 folds. With this setup, the dataset was divided into 10 subsets, and the model training and testing were performed iteratively over these folds. This procedure helps mitigate biases and provides a more reliable estimate of the performance.

Under these rigorous evaluation conditions, the model demonstrated a remarkable precision (see more in Section V). In Figure 6 the confusion matrix for the "Normal vs Abnormal" model is showed.

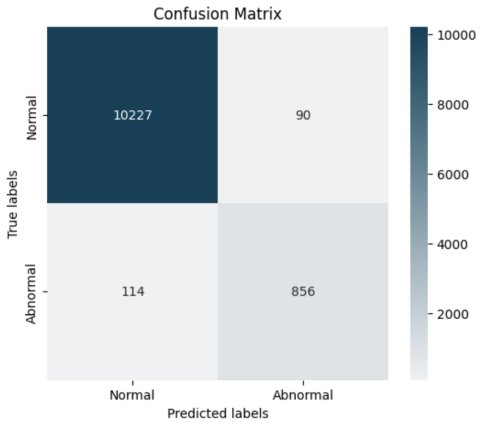

Fig. 6. Confusion matrix for the "Normal vs Abnormal model"

F. Bagging model

In alignment with the paradigm established for the normal versus abnormal model detailed in the previous section, a

specialized model was developed exclusively for the categorization of abnormal beats, aiming to differentiate between ventricular and supraventricular photoplethysmogram (PPG) beats. Regrettably, the model yielded unsatisfactory outcomes, indicating significant challenges in achieving effective discrimination between abnormal beats.

For this reason, a novel approach was adopted, specifically leveraging a technique called bagging (Breiman, 1996 [2]), short for *Bootstrap Aggregating*, which is an ensemble learning technique designed to enhance the stability and performance of machine learning models. The fundamental concept involves training multiple instances of the same base model on different subsets of the training data, which are created through random sampling with replacement, known as bootstrapping.

When predicting the class of a new input signal, each individual model contributes its prediction, and the ensemble decision is determined through a majority voting mechanism. This entails aggregating the class predictions from each model and selecting the class that attains the majority of votes. The application of bagging is particularly beneficial when dealing with complex and high-variance models as in this case.

In this scenario, 10 base models were defined with the Categorical Cross-Entropy loss function, aimed at classifying signals into one of three classes: 'N', 'S', 'V'.
In Figure 7 the confusion matrix for the "Bagging" is presented.

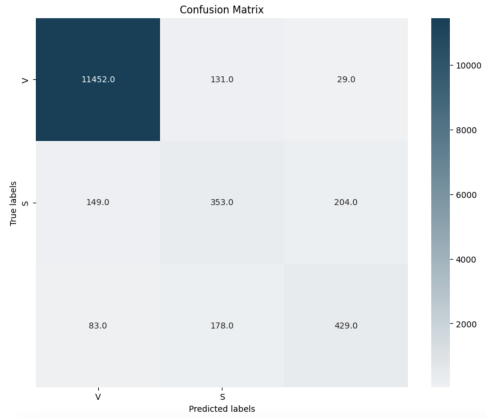

Fig. 7. Confusion matrix for the "Bagging" model

In Section V these results will be further explained using confidence intervals.

To ensure the reliability and consistency of the results, even in this case, a 10 folds cross-validation approach was incorporated into the methodology. This validation process took a considerably amount of time since for each test fold 10 models had to be trained (100 in total).

V. Evaluation

In evaluating the models, confidence intervals for error, precision, and recall for each considered class were computed using cross-validation scores.

A. Confidence Intervals for "Normal vs Abnormal" Model

In Table II, the confidence intervals for the first model are presented. Notably, tight intervals with a high mean characterize the 'N' class, while larger intervals with a lower mean describe the 'A' class. This difference may be attributed to class imbalance in the dataset. Despite these variations, the model consistently demonstrates strong performance across all cross-validation folds, indicating robust generalization capabilities.

TABLE II
Confidence Intervals for the "Normal vs Abnormal Model".

	Inf	**Mean**	**Sup**
Error class N (%)	0.0066	0.0099	0.01331
Error class A (%)	0.1460	0.2300	0.3141
Precision class N	0.9712	0.9787	0.9862
Precision class A	0.8590	0.8842	0.9095
Recall class N	0.9866	0.9900	0.9933
Recall class A	0.6858	0.7699	0.8539

B. Confidence Intervals for "Bagging" Model

In Table III the confidence intervals for the second model are presented. Similar to the findings for the first model, a notable difference is observed between the classes 'N' and both 'V' and 'S.' Specifically, the model exhibits proficiency in distinguishing between 'normal' and 'abnormal' samples but struggles to discriminate between different abnormalities in heartbeats. This challenge could be attributed to the under-representation of these abnormal classes compared to the normal class or the similarity between the two types of anomalies. Additionally, it is observed that the model's performance diminishes when evaluating ventricular or supraventricular samples, particularly when processing data from new patients in each fold of the cross-validation. This drop in performance may indicate several potential issues:

- The chosen model may not be suitable for effectively addressing this classification problem;
- Overfitting of the model to the training data, leading to a lack of generalization on unseen data. Interestingly, this was not evident during the training phase when observing the loss plot;
- Indistinguishability between the morphology of ventricular and supraventricular hearthbeats;
- The shape of ventricular and supraventricular beats may strongly depend on individual patient characteristics;

Further investigations might be needed in order to discover a proper solution for the classification of ventricular and sopraventricular beats in PPG signals.

TABLE III
Confidence Intervals for the "Bagging" Model.

	Inf	**Mean**	**Sup**
Error class N (%)	0.0050	0.0292	0.05336
Error class V (%)	0.3199	0.4268	0.5337
Error class S (%)	0.2915	0.3867	0.4819
Precision class N	0.9856	0.9887	0.9919
Precision class V	0.6010	0.6776	0.7542
Precision class S	0.3195	0.4214	0.5232
Recall class N	0.9466	0.9707	0.9949
Recall class V	0.4662	0.5731	0.6800
Recall class S	0.5180	0.6132	0.7084

In Figure 9 and Figure 10 in the Appendix A all the confidence intervals are plotted in a summary plot.

C. The Metric

In order to evaluate the overall performance of the bagging model, we introduce the *Weighted Precision Metric*. This metric is designed to consider the skewness of the dataset and compute the model's performance by taking into account the numerosity of each class in the dataset used as test set. The Weighted Precision ($wP(x)$) is calculated as:

$$wP(x) = \frac{\sum_{c \in C} n_c \cdot P_c}{\sum_{c \in C} n_c} \tag{1}$$

where n_c represents the numerosity of class c in the set of classes C, and P_c is the precision of the model classifying class c. The overall performance, computed for all three classes, scored 0.9471 using the mean precision obtained through cross-validation.

It's important to note that the high skewness of the dataset, particularly in the 'N' class, strongly influences this performance. Consequently, the metric has been computed solely for the under-represented classes ('V' and 'S'), resulting in a score of 0.5736. This score is well supported by the confidence intervals computed in Section V-B.

VI. Conclusions

Regarding the classification of PACs and PVCs using PPG signals, the model presented in Section IV seems to be robust in discriminating between "normal" and "abnormal" heartbeats, while struggles in identifying correctly ventricular or sopraventricular examples. However, the obtained results might derive also from the skewness in the representativeness of the dataset, which was taken into account partially. Further works may exploit oversampling techniques over the under-represented classes in order to augment the dataset. Moreover, the autoencoder used to detect artifacts may be not optimal for this specific clinical problem (see Section III-A).

Nevertheless, the integration of features regarding the morphology of the beats seems to increase the overall performance of the model, but future investigation could lead to the discovery of optimal statistical quantities specific for the problem.

Appendix

Appendix A

1) Figure 8 shows the overall pipeline that have been implemented to solve the task.

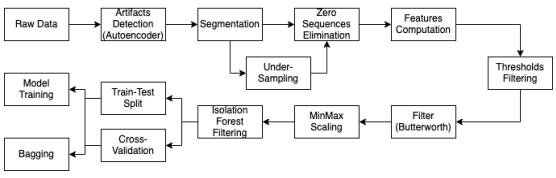

Fig. 8. The pipeline

2) Figure 9: confidence intervals for the model "Normal vs Abnormal".

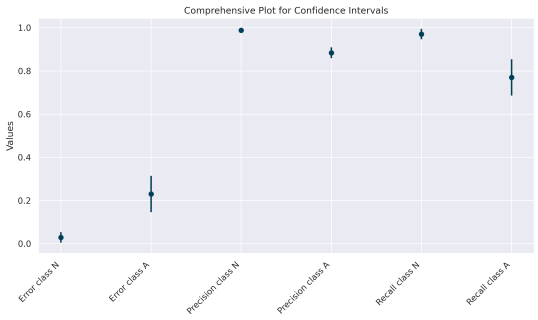

Fig. 9. Confidence intervals for the model "Normal vs Abnormal"

3) Figure 10: confidence intervals for the model "Normal vs Ventricular vs Sopraventricular".

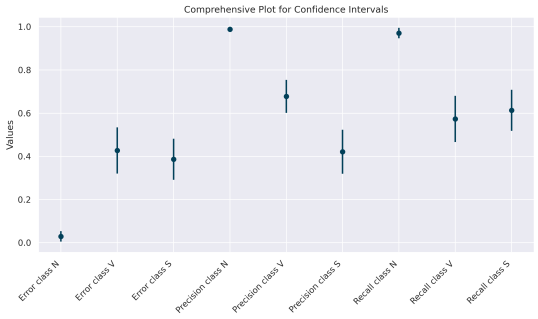

Fig. 10. Confidence intervals for the model "Normal vs Ventricular vs Sopraventricular"

References

[1] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: (2019). arXiv: 1907.10902 [cs.LG].

[2] Leo Breiman. "Bagging Predictors". In: Machine Learning 24.2 (1996), pp. 123–140.

[3] Sandipan Dhar et al. "Effortless detection of premature ventricular contraction using computerized analysis of photoplethysmography signal". In: Sādhanā 47.1 (2022), p. 28.

[4] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: (2015). arXiv: 1512.03385 [cs.CV].

[5] Dominique Makowski et al. "NeuroKit2: A Python toolbox for neurophysiological signal processing". In: Behavior Research Methods 53.4 (Feb. 2021), pp. 1689–1696. DOI: 10.3758/s13428-020-01516-y. URL: https://doi.org/10.3758%2Fs13428-020-01516-y.

[6] Pascal Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: ICML '08 (2008), pp. 1096–1103. DOI: 10.1145/1390156.1390294. URL: https://doi.org/10.1145/1390156.1390294.

[7] Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". In: (2018). arXiv: 1807.06521 [cs.CV].