Deep Learning based ECG Heartbeat Classification

PEDRO OSÓRIO[1]

[1] *pedro.louro@mail.polimi.it; Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal*

June 19, 2022

There has been studies suggesting that premature atrial complexes (PAC) are related to atrial fibrillation (AF) and cardiac tissue deterioration. In turn, premature ventricular complexes (PVC) have been linked with cardiomyopathy and ventricular dysfunction. Automated tools for the detection of these instances on ECG signal are useful for estimating their burden on a patient a better understand their impact on patient health and electrophysiology. In this work a deep learning based tool is presented for heart beat classification into normal (N), ectopic supraventricular (S) and ectopic ventricular (V) instances. Three different convolutional neural network (CNN) architectures were experimented with and the best performing one was selected via a 3-fold cross validation test. To train these models a dataset of 2-lead ECG signal from 105 patients with previously annotated R peaks was preprocessed and used. Segmentation of the beats consisted in selecting the patient median RR time interval to the left of the R peak and a percentage of it to the right. Multiple percentages were experimented with, concluding the best one to be 0.5. Approaches to deal with class imbalance were explored. The final model's attained a F1-score 0.995, 0.961 and 0.876 for classes N, V and S, respectively, on the test set.

INTRODUCTION

Heart diseases have been the main cause of death worldwide for the past two decades [1]. Ensuring accurate detection and diagnosis of these diseases will undoubtedly contribute to higher human life expectancy and well-being. There are several methods and techniques that can be used for detecting and monitoring cardiovascular diseases, of them electrocardiogram (ECG) is by far the most extensively used due to its affordability and convenience. ECG can depict changes in the depolarisation patterns in the myocardium which consist in valuable information for heart disease diagnosis.

In this paper the aim is the identification of premature atrial complexes (PAC) and premature ventricular complexes (PVC). In a patient with normal heart electrophysiology, each heart beat is initiated by an electrical pulse generated by the sinus node which will then propagate through the rest of the myocardium. However, premature heart beats can arise from ectopic pacemaking tissue which are located in other regions of the heart separated from the sinus node. ECG allows us to distinguish between heart beats stemming from ectopic pacemaking tissue within the atria and within the ventricular tissue as they give rise to signal with distinct morphology (PAC and PVC, respectively). The occurrence of these ectopic heart beats has been always considered benign but recently they have been linked to some diseases when frequent. PAC has been associated with high risk of developing atrial fibrillation (AF) and stroke [2] [3] [4].In fact, PAC have been linked with the first time appearance of AF [5] and also studied as possible measure of cardiac tissue deterioration [6]. In turn, PVC has been paired with ventricular dysfunction and can induce cardiomyopathy [7]. PVC has also been studied to be a possible symptom of myocarditis [8]. Adding to that, it is consensual that there is a lot more to know about what originates these ectopic beats and the impact that might have on human health.

Being able to detect these ectopic beats, distinguish between them and identifying their burden is then an important task that, considering PAC, could potentially contribute to the early detection of AF and consequently diminish the incidence rate of cryptogenic strokes as well as improve the performance of arrhythmia detectors [9]. In turn, identifying PVC burden is relevant as being a marker of ventricular dysfunction and cardiomyopathy it can allow us to track the evolution of these pathologies before and after interventions like catheter ablation or pharmacological suppression [8]. Adding to that, the detection of these instances would pave the way to the extension of our knowledge about them, most importantly about their impact on cardiac electrophysiology.

Detecting these complexes, however, is usually done manually and by visual inspection of the ECG signal, which due to its complexity, must be done by experienced doctors. Nevertheless, despite the acquisition of ECG being well standardised, their human interpretations can vary widely even within professionals with higher expertise. Moreover, due to the non-stationary

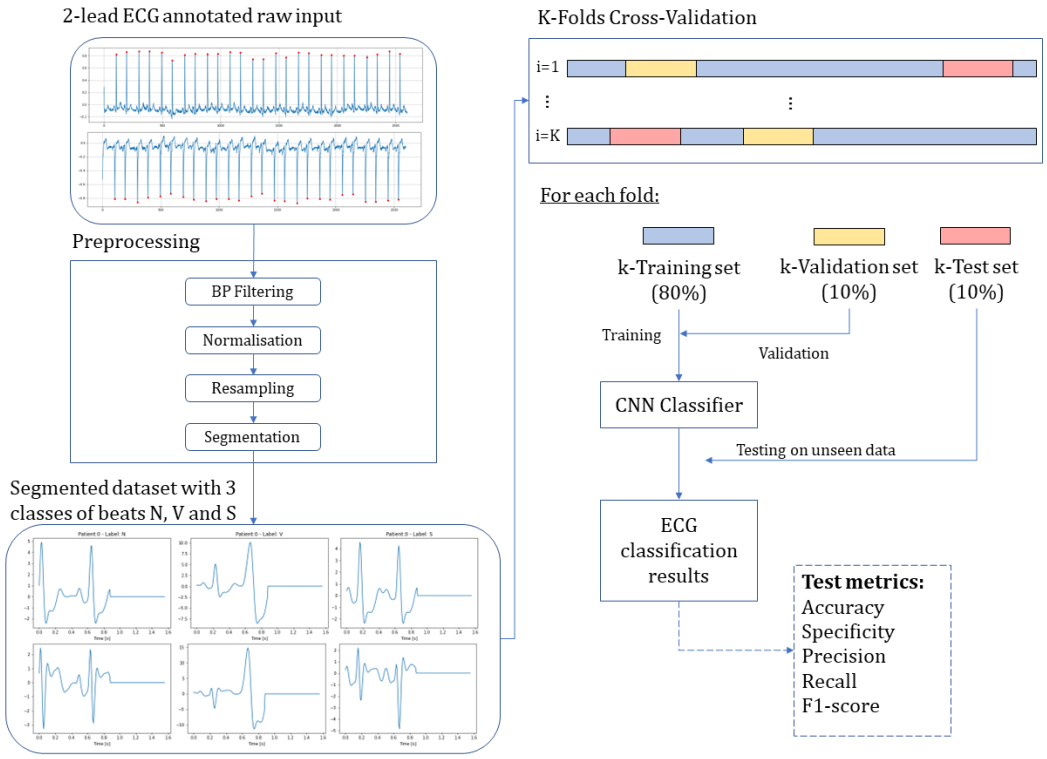

Fig. 1. The general outline of the method implemented and presented in this paper.

and non-linear nature of this signal, these instances emerge randomly along time making this method quite an challenging and time-consuming task, especially when talking about long-term acquisitions. Therefore, automatic approaches to this problem have been emerging and gaining increasing importance.

Computer interpretations of this data have been used but they rely on predetermined rules and are limited by the feature recognition algorithms in place resulting in interpretations that do not always capture the complexities and nuances contained in the ECG [10].

In past decades, many popular machine learning methods have been proposed to address this issue of ECG data classification. With respect to PAC and PVC detection methods like random forest algorithm [9] and decision tree [11] have been trained.

Within these machine learning approaches, deep learning methods, in particular Artificial Neural Networks (ANNs), stand out due to their impressive ability to learn from the ever increasing ECG datasets. As opposed to other machine learning approaches, ANNs have the ability to learn and extract meaningful patterns from complex raw data, not being limited by handcrafted features that can compromise the model's generalisation ability. Convolutional Neural Networks (CNNs) are the most used type of ANN with ECG data having been successfully employed for ventricular and supraventricular ectopic beat detection in [12], in [13] for classification into 9 different classes of ECG-based rhythms and in [14] for atrial fibrillation detection. Residual CNNs have also been used within this scope for for ventricular and supraventricular ectopic beat detection [15].

In this paper, a comparative study between various types of CNNs models for ventricular (PVC), supraventricular (PAC) and normal beat classification will be presented. Further detail will be provided regarding the ECG signal processing and segmentation as well as about the different architectures and training hyperparameters. The best performing model will then be selected from all the trained ones.

MATERIALS AND METHODS

The general outline of the preprocessing steps applied to the dataset, the split performed for the training of the several classifiers presented in this paper and the metrics used is depicted on Figure 1 and will be described with further detail in the next subsections.

Dataset

The dataset used in this work is comprised of 2-lead ECG acquisitions from a pool of 105 different subjects each with a duration of 30 minutes. This dataset came annotated with the locations of all R peaks which were labelled with their respective classes, meaning if they correspond to either a normal heartbeat (N), a supraventricular heartbeat (S) or a ventricular one (V).

Data Preprocessing

Filtering

The first step of the preprocessing pipeline consisted of band pass filtering the raw ECG signals between 1-35 Hz eliminating the frequency components that corresponded to the powerline and high frequency noise as well as hindering the lower frequency baseline wander typical of ECG. The filter employed was a 3rd order Butterworth band pass filter.

Normalisation

Each subject's ECG signal went through Z-score normalisation ensuring that its mean value is at zero and standard deviation at 1. This will guarantee that the signal collected from the different

subjects is comparable. Additionally, some of the ECG signals provided were acquired at different sampling frequencies so to ensure uniformity in the dataset, every signal was resampled to 128 Hz and their R-peak annotations were updated accordingly.

Segmentation

Only after these steps these signals are segmented into various beats based on the location of the annotated R-peaks using an approach inspired by [12]. The median R-R time interval (med_{RR}) is computed for each subject and then for each R-peak a section of the signal with a length of $(1 + \alpha)$ times said value is selected. This interval is not centred around the R-peak as the selection encompasses the med_{RR} seconds before the R-peak and the $\alpha * med_{RR}$ seconds after the peak. Since this method generates different length segments according to the subject's median heart rate, all the final crops are padded to a predefined fixed length. Multiple α values have been experimented with in order to understand which type of segmentation allowed better results.

Under the assumption that most of the beats are normal, segmenting based on the median R-R time interval of a subject might contribute to making normal and premature segments more distinct from each other as the previous QRS complex will tend to appear in the segments of premature beats and not in normal ones. This will hopefully ease the learning process of the networks and increase their performance at this classification task. See Figure 2 for a clear schematic of the segmentation method.

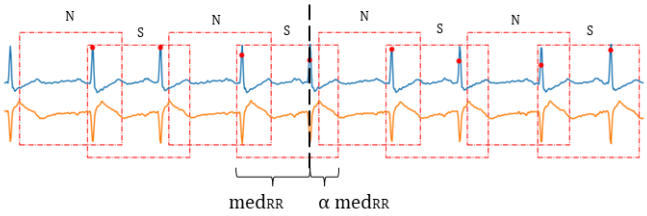

Fig. 2. ECG segmentation scheme depicted on both leads' signal. med_{RR} stands for median R-R time interval and α is a factor. Each segment is labelled.

Split

After segmenting the ECG signal, the obtained dataset is comprised of a 244204 heart beat samples labelled with their respective class. Of them only approximately 3.19 % are labelled as ventricular ectopic beats and 3.93 % as atrial ectopic beats, which makes our data quite unbalanced (see Table 1).

To train and evaluate the models that will be described later on it is common to perform a random split of the dataset into training, validation and testing data. This is called hold out cross validation and allows for models to be tested with never seen data from the test subset and their training monitored with the validation subset. A more time consuming but unbiased approach is K-fold cross validation. In relation to a simple hold-out, this technique allows for more unbiased comparison as our results are the average of the performance of the models in three different splits as opposed to only one. Both these techniques were used in this work depending on the analysis at hand. The split percentages were 80-10-10 and the number of fold was 3.

In situations where the number of examples of each class are more or less the same the splits can be done randomly from the full bulk of samples but in this case, due to the imbalance of the dataset, the splits were done in stratified fashion, meaning class by class, to ensure that every subset has the same distribution of samples per class as the full dataset.

Table 1. Distribution of the dataset and subsets in different categories after preprocessing and segmentation.

Category	N	V	S	Total
Train	181444	6235	7684	195363
Validation	22681	779	961	24421
Test	22680	780	960	24420
Total	226805	7794	9605	244204

SMOTE

As the dataset in use is quite unbalanced, a Synthetic Minority Oversampling Technique (SMOTE) [16] was implemented and experimented with. This method showed good results in [14] and works by selecting close elements of the minority class in the feature space and generating new samples by sampling the line that connects them. As it is stated in the original paper, a combination of SMOTE and undersampling of the majority class should be used for better results so that is what was done. Essentially, the total number of samples in the training set was kept the same by oversampling the minority classes by their class weights (V:10.44 and S:8.478) and undersampling the majority class by its class weight (N:0.358).

Models Tested

Architectures

Three types of CNN architectures were tested in this work and their description will follow.

The VGG classical architecture was adapted in two different ways to address this classification problem. Both versions adapt the original network for 1D inputs by making the convolution layers within its feature extraction block 1D but keep the original small kernel size of 3 units. One of the versions, **VGG1**, maintains the original depth of the classic VGG, having the same number and configuration of convolutional and MaxPooling layers (13 and 5, respectively) as well as filters in the feature extraction block and the same 2 layers dense block only with the last one altered to adapt to our number of classes. The max pooling layers are of size and stride 2. The layer connecting the previous two blocks is changed to from a Flatten to a GlobalMaxPooling1D layer to reduce complexity and to make the network more invariant to translations in the input. A dropout layer is also included between the two dense layers to reduce overfitting.

VGG2 is the other VGG inspired network used in this work and was proposed by [12] for the same task of ECG signal classification. As opposed to VGG1 and the classic VGG architecture, this version is less deep having 9 convolutional layers instead of 13 and 4 max pooling layers instead of 5. VGG2 maintains the original Flatten layer turning the 2 dimensional feature maps into a 1 dimensional vector that can be fed into the classification block. Similarly to VGG1, the same dense block is used, only with the first layer number of units decreased from 512 to 30. For this network both ReLu and Elu activation functions were experimented with, as using the latter avoids the ReLu dying

neuron problem. This is when the contribution of a neuron is consistently nullified due to ReLu zeroing negative inputs, not allowing the flow of the gradient and thus learning. To hinder overfitting, dropouts were also introduced after the second, fourth, sixth and ninth convolutional layer at 0.1 rate. This less deep and less complex network was experimented with to test the hypothesis of it better adapting to our smaller size input than a more complex model with a larger amount of trainable parameters and chance of overfitting.

The last CNN architecture experimented with was one based on the residual learning concept. The **ResNet** [17] is a state of the art CNN architecture that uses residual blocks for most convolutions allowing the network to train deeper without vanishing gradients. In a normal CNN the weight updates become infinitely smaller as we go deeper, due to repeated multiplications during the backpropagation of the gradient, thereby making deeper layers harder to train. The ResNet avoids this problem by using convolutional blocks with skip connections. The particular ResNet used consisted of 8 residual blocks.

Training

The categorical cross entropy was chosen as loss function for training the aforementioned models, as it provides a good measure of how distinguishable two discrete probability distributions, in this case true labels and predicted ones, are from each other. Adam was the chosen optimiser algorithm with a batch size of 64 and the learning rate was halved when the validation results stop improving, has it has been proved to provide better results.

To avoid overfitting, early stopping was used to terminate the training when the validation results worsen and select the best epoch model according to a predefined metric. Even though the loss used was categorical cross entropy, the validation metric being monitored for both learning rate reduction and early stopping was the macro averaged F1-score. This score showed to be a better measure of overfitting, as in some situations, even though for the majority of samples the model provides very low loss, when a few outlier samples with larger loss are considered the batch overall loss can increase greatly without it necessarily reflecting negatively on the validation F1 scores. This is due to the fact that cross-entropy is unbounded, meaning it can take values from 0 to infinity [18]. F1-score performance is given priority in relation to the cross entropy as it is defined based on the objective metrics precision and recall.

As an attempt to address the imbalance in the dataset, class weights were used to weight the contribution of each class instance to the loss based on their frequency in the training set.

The training and evaluation of the models was achieved using the TensorFlow open-source software library [19] for Python and the Google Colaboratory Notebook with the code is provided along side this report.

Evaluation Metrics

In order to evaluate the performance of the trained models on the test subset, some classification metrics were computed both for each class separately and also considering all classes. The four criteria were accuracy, specificity, recall (or sensitivity), precision (or positive predictive value) and the f1-score, which is the harmonic average of the last two indicators. See their respective formulas below:

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad \textbf{(1)}$$

$$Specificity(\%) = \frac{TN}{TN + FP} \times 100 \quad \textbf{(2)}$$

$$Recall(\%) = \frac{TP}{TP + FN} \times 100 \quad \textbf{(3)}$$

$$Precision(\%) = \frac{TP}{TP + FP} \times 100 \quad \textbf{(4)}$$

$$F1\text{-}score = \frac{Recall \times Precision}{Recall + Precision} \quad \textbf{(5)}$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. Note that the F1-score were computed based on the Precision and Recall values not multiplied by 100, therfore their range o fvalues will be between 0 and 1.

To obtain a single measure of each model's performance across the 3 classes, the macro F1-score was computed. This metric is the mean value of the F1-score across all classes, which is fitting to evaluate our model while not giving any extra importance to the performance at any class. For monitoring these metrics during training on the validation set, custom metric functions had to be implemented and the code is available on the provided notebook.

$$Macro\text{-}F1\text{-}score = \frac{F1\text{-}score_N \times F1\text{-}score_V \times F1\text{-}score_S}{3} \quad \textbf{(6)}$$

RESULTS

To assess if the use of class weights would improve the performance of the models, the VGG2 network was trained both with and without them. A single hold out cross validation test yielded the results on Table 2 which depict better performances on the training without the class weights. So for then on class weights were discarded.

Table 2. Performance of VGG2 model with and without class weights on a hold out cross validation test. CW stands for class weights.

Model		F1-Score			Macro-F1-Score
		N	V	S	
VGG2	No CW	0.989	0.954	0.793	0.912
	CW	0.986	0.947	0.752	0.895

Still as an attempt to address the class imbalance, a SMOTE dataset was created and the VGG1 and VGG2 models were trained both on it and in the original unbalanced dataset. Their performances were evaluated on a single hold out cross validation test, whose results can be seen on Table 3. The best over all results were obtained without using SMOTE, so from then on this technique was discarded.

The original implementation of the VGG2 network in [12] employs the elu activation function on all layers instead of the most commonly used one ReLu. In this work both were experimented with, reaching the conclusion that ReLu better fits to our specific task and dataset. The results of the 3-fold cross validation test can be seen on Table 4.

Once defined the best activation function for VGG2 and discarded the SMOTE and class weights methods, a comparison of the 3 models performance was put through. The 3-fold cross

Table 3. Performance of VGG2 and VGG1 models with and without using SMOTE on a hold out cross validation test.

Model		F1-Score			Macro-F1-Score
		N	V	S	
VGG1	Original	0.995	0.951	0.864	0.937
	SMOTE	0.992	0.955	0.832	0.926
VGG2	Original	0.99	0.95	0.864	0.936
	SMOTE	0.991	0.952	0.819	0.921

Table 4. Performance of VGG2 model using Elu and Relu activation functions on a 3-fold cross validation test.

Model		F1-Score			Macro F1-Score
		N	V	S	
VGG2	Elu	0.992 ± 0.002	0.948 ± 0.007	0.849 ± 0.003	0.931 ± 0.003
	Relu	0.993 ± 0.002	0.947 ± 0.005	0.862 ± 0.007	0.935 ± 0.003

validation results are shown on Table 5 letting us conclude that the best performing model is VGG2.

Once the best architecture was selected, 3 types of heart beat segmentation were experimented with to determine the optimal one. The VGG2 architecture was trained with a data set of samples created with a α factor of 0.4, 0.5 and 0.6 and padded to a final length of 200, 210, 225 samples, respectively. The results of the 3-fold cross validation test can been seen on Table 6. Note that all the previous results were obtained on a data set of sample length 200 and α at 0.4.

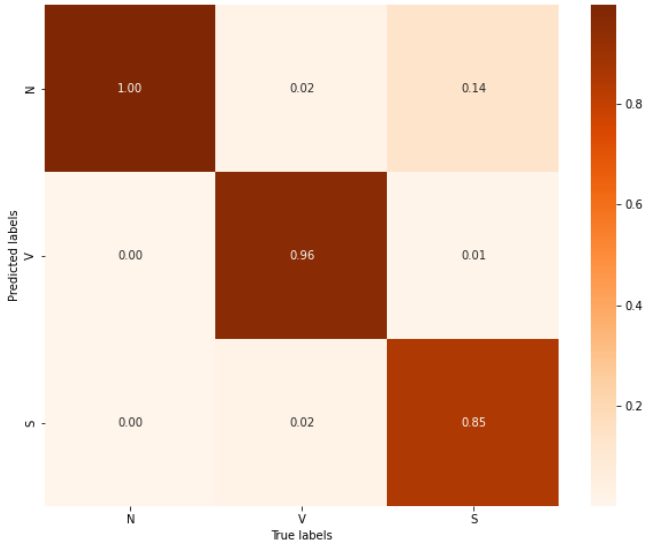

Fig. 3. Confusion matrix depicting the performance of the best model on the test set on the best fold.

DISCUSSION AND CONCLUSION

From the results presented above we are able converge on a final best model with VGG2 architecture, no SMOTE nor class weights and using a segmentation factor α of 0.5. This model's full test results can be seen on Table 7, the confusion matrix of the best fold can be observed on Figure 3 and the summary of its architecture can be seen in the Appendix's Figure 4.

Due to the larger number of samples of normal heartbeats, it was expected that the models would perform better at classifying these instances, which it did. However, it is relevant to note that, despite existing more instances of supraventricular beats than ventricular ones in the training set, all models consistently performed better at classifying correctly instances of class V than S. In fact, looking at the confusion matrix at Figure 3, it is

Table 5. Performance of VGG1, VGG2 and ResNet models on a 3-fold cross validation test.

Model	F1-Score			Macro F1-Score
	N	V	S	
VGG1	0.993 ± 0.002	0.948 ± 0.003	0.858 ± 0.006	0.933 ± 0.003
VGG2	0.995 ± 0.000	0.955 ± 0.003	0.873 ± 0.003	0.941 ± 0.001
ResNet	0.993 ± 0.002	0.956 ± 0.005	0.869 ± 0.002	0.939 ± 0.002

Table 6. 3-fold cross validation test performance of the VGG2 model trained on differently segmented datasets.

Model		F1-Score			Macro F1-Score
		N	V	S	
VGG2	$\alpha = 0.4$ $len = 200$	0.995 ± 0.000	0.955 ± 0.003	0.873 ± 0.003	0.941 ± 0.001
	$\alpha = 0.5$ $len = 210$	0.995 ± 0.000	0.961 ± 0.007	0.876 ± 0.001	0.944 ± 0.003
	$\alpha = 0.6$ $len = 225$	0.993 ± 0.002	0.954 ± 0.010	0.877 ± 0.002	0.943 ± 0.002

Table 7. 3-fold cross validation test performance of the best VGG2 model with the full test report.

VGG2 - Best Model		
N	V	S
Accuracy		
99.066 ± 0.038	99.748 ± 0.049	99.046 ± 0.018
Specificity		
92.452 ± 0.637	99.853 ± 0.047	99.593 ± 0.035
Precision		
99.42 ± 0.05	95.617 ± 1.345	89.621 ± 0.755
Recall		
99.574 ± 0.06	96.538 ± 0.456	85.66 ± 0.428
F1-Score		
0.995 ± 0.000	0.961 ± 0.007	0.876 ± 0.001
Macro-F1-Score		
0.944 ± 0.003		

clear that the worse F1-score seen for the S class stems from the missclassification of supraventricular instances misclassified as normal ones.

The main differences between ectopic beats and normal ones lie on their premature nature that disrupts the RR sequence and on the morphological distortion of the QRS. The fact is that this distortion is quite more distinct in PVC than in PAC (see Figure 1), which ultimately reflects on the model confusing S instances as normal ones. More precisely, the changes in QRS morphology in PAC are limited to small amplitude distortion on the p-wave, whereas in PVC the changes affect the whole QRS complex significantly.

It was found that using the full patient median RR interval to the left of the R peak and half of it to the right was the best input segmentation, providing not too much information ($\alpha = 0.6$) while also not too little ($\alpha = 0.4$). Given the robustness of the final model, it is fair to say that the segmentation technique employed is adequate, suggesting that the initial hypothesis of it allowing a better detection of the RR sequence disruption is correct.

The SMOTE, as implemented, showed not to provide good results. This upsampling could be introducing a patient-specific bias in the dataset by synthesising new samples mostly from samples of the same individual and thereby limiting the generalisation ability of the model. An additional Table 8 is provided on the Appendix section that clearly depicts how much more overfitted the SMOTE model is in relation to the one without it. This is shown by the large training-validation and validation-test f1-score variance that relate to the fact that the net learned features on training set that do not generalise to the other subsets.

It is also possible to say that the comparisons made on the basis of a single fold cross validation could be biased to the specific split used. Using a multiple fold cross validation test is always a better option especially when dealing with smaller datasets. The bias of this evaluation decreases for a larger number of folds but the decision of staying with only 3 was made for time management reasons. Increasing it is left as a suggestion for future works. Another suggestion to be better assess the model's robustness would be ensuring that samples from the patient are not used both on the test and training sets.

In the literature it is possible to find several preprocessing routines for ECG prior to beat classification. Alternatively to the here implemented band pass filtering for noise and artefact removal, the Discrete Wavelet Transform for signal denoising could also be a good bet. It has a high time and frequency resolution and was also successfully used in [15] and [13].

In the end, this work provides a comparison between various model architectures and input ECG segmentation converging on a optimal combination of these with good overall accuracy, specificity, precision and recall. It offers a valuable automated tool for a fast and automated classification of supraventricular, ventricular and normal heart beats, which contributes to a better understanding of the impact of ectopic beats on patient health. The gap to a even better performance could potentially be closed training on a larger, more diverse and balanced dataset. Adding to that, a random hyperparameter search to tune to the best dropout rate and batch size could be done in the future to improve results.

REFERENCES

1. W. H. Organization, "World health statistics 2019: Monitoring health for the sdgs, sustainable development goals„" (World Health Organization, Geneva, Switzerland, 2019).
2. Z. Binici, T. Intzilakis, O. W. Nielsen, L. Køber, and A. Sajadieh, "Excessive supraventricular ectopic activity and increased risk of atrial fibrillation and stroke," Circulation **121**, 1904–1911 (2010).
3. B.-t. Huang, F.-y. Huang, Y. Peng, Y.-b. Liao, F. Chen, T.-l. Xia, X.-b. Pu, and M. Chen, "Relation of premature atrial complexes with stroke and death: Systematic review and meta-analysis," Clin. cardiology **40**, 962–969 (2017).
4. D. J. Gladstone, P. Dorian, M. Spring, V. Panzov, M. Mamdani, J. S. Healey, K. E. Thorpe, E. S. C. or Operations Committee, R. Aviv, K. Boyle *et al.*, "Atrial premature beats predict atrial fibrillation in cryptogenic stroke: results from the embrace trial," Stroke **46**, 936–941 (2015).
5. T. Thong, J. McNames, M. Aboy, and B. Goldstein, "Prediction of paroxysmal atrial fibrillation by analysis of atrial premature complexes," IEEE Transactions on Biomed. Eng. **51**, 561–569 (2004).
6. B. S. Larsen, P. Kumarathurai, J. Falkenberg, O. W. Nielsen, and A. Sajadieh, "Excessive atrial ectopy and short atrial runs increase the risk of stroke beyond incident atrial fibrillation," J. Am. Coll. Cardiol. **66**, 232–241 (2015).
7. T. S. Baman, D. C. Lange, K. J. Ilg, S. K. Gupta, T.-Y. Liu, C. Alguire, W. Armstrong, E. Good, A. Chugh, K. Jongnarangsin *et al.*, "Relationship between burden of premature ventricular complexes and left ventricular function," Hear. rhythm **7**, 865–869 (2010).
8. B. Gorenek, J. D. Fisher, G. Kudaiberdieva, A. Baranchuk, H. Burri, K. B. Campbell, M. K. Chung, A. Enriquez, H. Heidbuchel, V. Kutyifa *et al.*, "Premature ventricular complexes: diagnostic and therapeutic considerations in clinical practice," J. Interv. Cardiac Electrophysiol. **57**, 5–26 (2020).
9. G. García-Isla, L. Mainardi, and V. D. A. Corino, "A detector for premature atrial and ventricular complexes," Front. Physiol. **12** (2021).
10. K. C. Siontis, P. A. Noseworthy, Z. I. Attia, and P. A. Friedman, "Artificial intelligence-enhanced electrocardiography in cardiovascular disease management," Nat. Rev. Cardiol. **18**, 465–478 (2021).
11. V. Krasteva, R. Leber, I. Jekova, R. Schmid, and R. Abächerli, "Classification of supraventricular and ventricular beats by qrs template matching and decision tree," in *Computing in Cardiology 2014,* (IEEE, 2014), pp. 349–352.
12. D. Zhang, Y. Chen, Y. Chen, S. Ye, W. Cai, and M. Chen, "An ECG heartbeat classification method based on deep convolutional neural network," J. Healthc. Eng. **2021**, 1–9 (2021).
13. A. Darmawahyuni, S. Nurmaini, M. N. Rachmatullah, B. Tutuko, A. I. Sapitri, F. Firdaus, A. Fansyuri, and A. Predyansyah, "Deep learning-based electrocardiogram rhythm and beat features for heart abnormality classification," PeerJ Comput. Sci. **8**, e825 (2022).
14. S. Nurmaini, A. E. Tondas, A. Darmawahyuni, M. N. Rachmatullah, R. U. Partan, F. Firdaus, B. Tutuko, F. Pratiwi, A. H. Juliano, and R. Khoirani, "Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks," Futur. Gener. Comput. Syst. **113**, 304–317 (2020).
15. E. Jing, H. Zhang, Z. Li, Y. Liu, Z. Ji, and I. Ganchev, "ECG heartbeat classification based on an improved ResNet-18 model," Comput. Math. Methods Medicine **2021**, 1–13 (2021).
16. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," J. artificial intelligence research **16**, 321–357 (2002).
17. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* (2016), pp. 770–778.
18. ahstat (https://stats.stackexchange.com/users/155499/ahstat), "Good accuracy despite high loss value," Cross Validated. URL:https://stats.stackexchange.com/q/281651 (version: 2017-05-25).
19. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker,

V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," (2016).

APPENDIX

Table 8. Variance of F1-scores between training and validation datasets on the VGG1 model, meaning the absolute difference between the train and validation metrics and validation and test metrics.

Model		F1-Score Class S		Macro F1-Score	
		Variance	Val-Test Variance	Variance	Val-Test Variance
VGG1	Original	0.135346	0.002648	0.062710	0.000115
	SMOTE	0.181564	0.034122	0.081725	0.006156

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_18 (Conv1D)           (None, 208, 16)           112

conv1d_19 (Conv1D)           (None, 206, 16)           784

max_pooling1d_8 (MaxPooling  (None, 103, 16)           0
1D)

dropout (Dropout)            (None, 103, 16)           0

conv1d_20 (Conv1D)           (None, 101, 32)           1568

conv1d_21 (Conv1D)           (None, 99, 32)            3104

max_pooling1d_9 (MaxPooling  (None, 49, 32)            0
1D)

dropout_1 (Dropout)          (None, 49, 32)            0

conv1d_22 (Conv1D)           (None, 47, 64)            6208

conv1d_23 (Conv1D)           (None, 45, 64)            12352

max_pooling1d_10 (MaxPoolin  (None, 22, 64)            0
g1D)

dropout_2 (Dropout)          (None, 22, 64)            0

conv1d_24 (Conv1D)           (None, 20, 128)           24704

conv1d_25 (Conv1D)           (None, 18, 128)           49280

conv1d_26 (Conv1D)           (None, 16, 128)           49280

max_pooling1d_11 (MaxPoolin  (None, 8, 128)            0
g1D)

dropout_3 (Dropout)          (None, 8, 128)            0

flatten_2 (Flatten)          (None, 1024)              0

dense_4 (Dense)              (None, 30)                30750

dense_5 (Dense)              (None, 3)                 93

=================================================================
Total params: 178,235
Trainable params: 178,235
Non-trainable params: 0
```

Fig. 4. Best model VGG2 summary.