Comparison of N-BEATS and SotA RNN architectures for heart dysfunction classification

B Puszkarski[1], K Hryniów[2] and G Sarwas[2]

[1] Faculty of Electrical Engineering, Warsaw University of Technology, Koszykowa Street 75, 00-662 Warsaw, Poland

[2] Institute of Control and Industrial Technology, Warsaw University of Technology, Koszykowa Street 75, 00-662 Warsaw, Poland

E-mail: [1] puszkarski.bartosz@gmail.com

E-mail: [2] krzysztof.hryniow@pw.edu.pl, grzegorz.sarwas@pw.edu.pl

December 2021

Abstract. The paper deals with the problem of prediction and classification of electrocardiogram (ECG) signals. This type of signal can be classified as time series in which recursive neural networks (RNNs) are used for solving. The most frequently used network for such a solution is the Long Short-Term Memory (LSTM) architecture.

This paper is an extension of study presented at the CinC conference as WEAIT team, which investigated the use of Neural Basis Expansion Analysis for Interpretable Time Series (N-BEATS) for ECG analysis. The main purpose of this work is to compare the results obtained by N-BEATS, together with the results that can be obtained, with various types of other SotA (state-of-the-art) recurrent neural network architectures like LSTM, LSTM with peepholes, GRU. In addition, a performance comparison (both accuracy and time needed) is conducted for a different number of electrocardiogram leads, as obtaining results with a reduced number of leads allows for arrhythmias detection and classification while using off-the-shelf wearable devices (Holter monitors, sport bands, etc.).

Keywords: ECG Classification, Cardiovascular diseases, Neural network, RNN, GRU, N-BEATS, LSTM

1. Introduction

One of the most common causes of death globally are still CVDs (Cardiovascular diseases). Detection and classification of cardiac arrhythmias are often done with the help of an electrocardiogram (ECG) signal representing the heart's electrical activity. A specialist can visually inspect the ECG waveform, but as CVD symptoms at earlier stages can occur only occasionally, continuous monitoring by wearable devices is used.

The widespread miniaturization of electronic devices, combined with high availability and low price, makes measuring heart rate or saturation an addition to

everyday devices, such as smartwatches. Therefore, it is worth asking to what extent the data collected from those devices can be helpful. Miniaturization of the current professional medical devices could increase the availability of heartbeat analyses similar to those obtained with the Holter monitor and improve heart disease diagnosis. The development of such devices and algorithms would limit the impact of the population's neglect in diagnosing cardiovascular diseases and improve the effectiveness of the therapy.

The most commonly used solutions for diagnosing cardiac problems are offline processing of stored signals, remote processing on cloud servers, and local execution on a wearable device. Offline processing is the oldest solution and allows for easy classification using the whole dataset but cannot be done for real-time detection problems. It is now commonly used in medicine for less intensive cases. Patients wear remote devices like Holter monitors during their daily routine, and data processing happens during the visit to a cardiologist. Cloud processing is becoming more common, allowing the power of cloud computing to process signals from devices online, but raises valid privacy and security issues. Local execution mitigates all those problems, allows for non-stop operation and real-time detection, regardless of network coverage, while retaining security. The main problem with such a solution is that the automatic ECG classification algorithm must keep its accuracy while being lightweight enough for less powerful processors used on wearable devices. This paper examines a solution that works on server processors, GPU units, and low-powered processors on wearable devices.

Algorithms based on morphological features and classical signal processing techniques were used before. Fixed solutions proved insufficient due to the ECG waveform's morphological characteristics' variance between patients and circumstances of measurement [1]. Because of that, deep-learning-based algorithms using recurrent neural networks (RNNs) and convolutional neural networks (CNNs) were proposed [1, 2, 3, 4]. ECG classification and prediction are based on the problem of time-series analysis. In that field, one of the most commonly used classifiers is the Long Storm-term Memory (LSTM) network. The same network was used as a basis for this study. Also, the 2020 PhysioNet / Computing in Cardiology Challenge showed that the most common solutions used for 12-lead classification were also CNNs and RNNs [5]. Neural Basis Expansion Analysis for Interpretable Time Series (N-BEATS) [6, 7] is one of the newest RNNs, used for forecasting time series.

[8] introduced versions of the LSTM and GRU algorithms, consisting of additional wavelet analysis and merged predictions from smaller models to lower the computational cost. In the study, [9] we examined for the 2021 PhysioNet Challenge the use of modified N-BEATS as a multi-label classifier for cardiac problems, and the study showed that while its results were sub-par, for a low number of electrocardiogram leads, it achieved acceptable results while maintaining low complexity that would allow to use it on a wearable device.

As an extension to the study [9] in this article, we present the comparison of N-BEATS and three common architectures (LSTM, LSTM with peepholes [10] and Gated

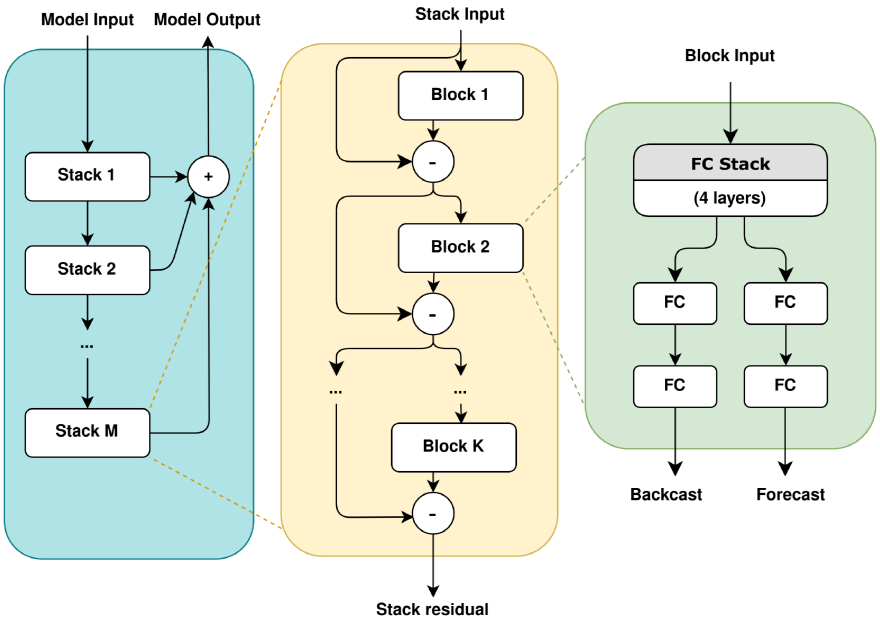

Figure 1. Model of N-BEATS network.

Recurrent Unit (GRU) [11]), modified with added two blended sub-networks and wavelet analysis. Furthermore, we present a performance comparison for a different number of electrocardiogram leads.

2. Methods

Time series forecasting is an important problem in the machine learning field. Statistical methods are still in use, despite the rapid development of machine learning algorithms. Hybrid solutions, using neural residual/attention extended LSTM stack with the classic Holt-Winters statistical model [12] with learnable parameters are becoming increasingly common. Article [6] introduced N-Beats architecture for interpretable time series forecasting (architecture model is in the Figure 1). The basic building block of this architecture is a multi-layer fully connected (FC) network with nonlinearities provided by activation function ReLU. It predicts basis expansion coefficients both forward (forecast) and backward (backcast). Blocks are connected into stacks using the doubly residual stacking principle, which may have layers with shared backcast and forecast blocks. Hierarchical fashion is used to aggregate forecasts, enabling a very deep neural network with interpretable outputs.

As the essential part of the signal for CVDs diagnosis is part around the R peak [8, 13], the R peak detection algorithm is used for signal segmentation. In the article, we used Pan-Tompkin's algorithm [14] for the R peak detection and segmentation process. Digitized ECG samples are segmented into a sequence of heartbeats containing precisely 0.7 seconds of an input signal (0.25 seconds before R peak and 0.45 seconds after R peak).

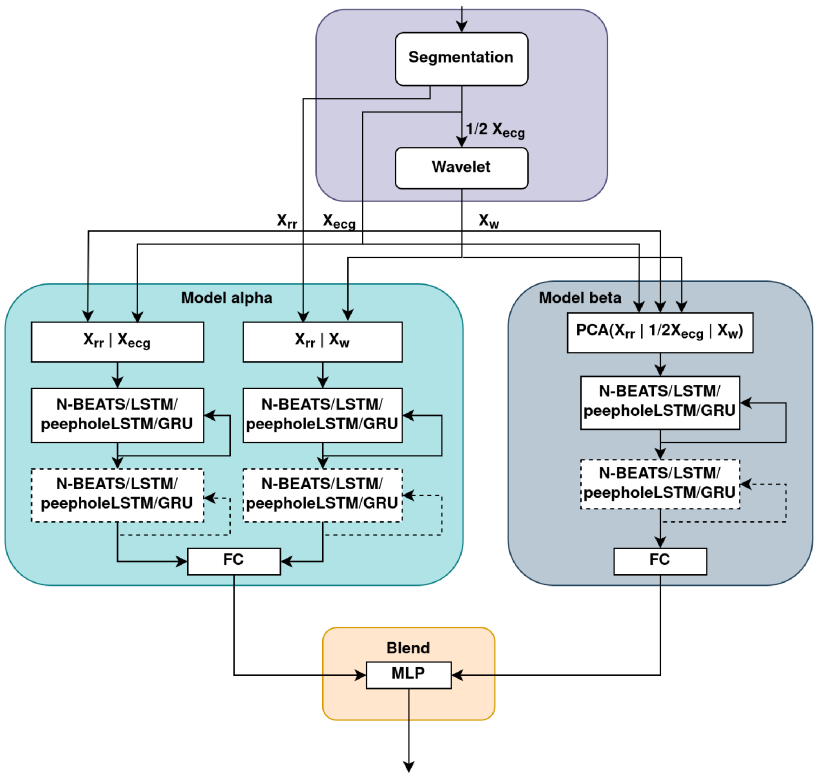

Figure 2. Model of classifier algorithm using LSTM / N-BEATS / GRU network.

This signal is denoted as X_{ecg} in Fig. 2. As in [8], additional information is obtained in the form of X_{rr} feature vector as part of Pan-Tompkin's algorithm's workings. The vector consists of three features of heartbeat: $X_{rr}^{(1)}$ is the last peak's interval, $X_{rr}^{(2)}$ is the next peak's interval and $X_{rr}^{(3)}$ is the average duration of five past and four next intervals. To allow the use of the algorithm in real-time, first-in-first-out (FIFO) memory is used for buffering ten ECG signals.

To capture time and frequency domain information, to downsampled (by a factor of two) digitalized ECG samples, type 2 Daubechies wavelet transform (db2) with four levels of decomposition is applied. The result is the second input vector $X_w = (A4, D4, D3, D2, D1)$.

All four algorithms use two models (donated as α model and β model) consisting of multiple parallel RNNs blended using a multi-layer perceptron (MLP) network with two hidden layers (Figure 2). The α model processes X_w and X_{ecg} inputs in separate branches consisting of parallel RNNs. Outputs of those branches are concatenated and used as input for a fully connected neural network layer (FC) to produce probabilities of all output classes. In the β model, the principal component analysis (PCA) is used on a downsampled version of X_{ecg} concatenated with X_w. The result is used as input for RNNs, and extracted features form the input for FC neural network layer. Results from both models are blended using MLP to obtain the final probabilities of output classes.

3. Experiments

3.1. Dataset

Our experiments were conducted on publicly shared training data sets, consisting of annotated twelve-lead ECG over 88,000 ECGs [15, 16, 17, 18, 19, 20]. For results comparison, we used cross-validation techniques – the whole dataset was divided using five folds for the training and testing datasets. Network weights were initialized using frozen seed to randomize the validation dataset for each training dataset to select a number of epoch for the training process. The number of epochs for a given training set was based on the early stopping technique and validation dataset. Then, we fixed obtained number of epochs and trained the model for all training data in this fold using network weights initialization for frozen seed. Model was tested by calculating average accuracy, F-measure, and challenge metric. As speed and low computational complexity are important for online processing on wearable devices, for each experiment execution times for one R peak were also noted.

3.2. Scenarios

To test the capabilities of chosen neural networks multiple experiments were conducted for each of four scenarios described below.

- Scenario A — In first scenario comparison between classic (one-branch) architecture and two-branch ($\alpha\beta$) architecture introduced in [8] and presented in Figure 2 was conducted for LSTM, GRU and N-BEATS. It it known that for LSTM neural network two-branch architecture achieves better results, so the objective of the experiment was to check if it is true for other networks and if this improvement holds also for score metric used in CinC challange.
- Scenario B — This scenario checks if unequal two-branch configurations, where in one branch network of larger size is used (dominant α or dominant β) works better then configuration with equal-sized branches. It is conducted on N-BEATS, GRU and LSTM networks for 2 leads only. For each network tests were conducted for 3 architecture configurations of varying sizes, each tested with 5 folds.
- Scenario C — In this scenario comparison of four different architectures (LSTM, LSTM with peepholes, GRU, N-BEATS) for different number of leads (2, 3, 4, 6, 12) is performed for the same size of networks. This allows to compare accuracy, score metric and speed between tested solutions.
- Scenario D — In final scenario tuning is performed for different networks, to determine what are the best results that can be obtained with each of them.

3.3. Results

The experiments were performed for signals consisting of 2, 3, 4, 6, and 12 ECG leads. For tests were taken only 26 classes, defined by the CinC challenge organizer, which

Table 1. Scenario A – comparison of LSTM, GRU and N-BEATS one- and two-branch architectures. Time is in milliseconds for one R peak.

Leads	LSTM		LSTM $\alpha\beta$		N-BEATS		N-BEATS $\alpha\beta$		GRU		GRU $\alpha\beta$	
	score	time	score	time	score	time	score	time	score	time	score	time
12			0.40	0.0036			0.27	0.0002			0.39	0.0037
6			0.38	0.0036			0.22	0.0002			0.39	0.0036
4			0.39	0.0027			0.27	0.0002			0.39	0.0027
3			0.39	0.0022			0.30	0.0002			0.39	0.0022
2			0.38	0.0017			0.34	0.0002			0.38	0.0017

described 30 heart diseases. The main goals of the research were to check the possibility of using the N-Beats model in the classification process, comparing the capacities of this architecture with the LSTM architecture, determining the required number of leads for the correct diagnosis of as many diseases as possible, and determining the computational complexity of the analyzed algorithms and their inference times. It was done to determine a solution that will allow the diagnosis of CVDs while retaining the speed and low computational capacity needed to use it for real-time detection on an off-the-shelf wearable device.

For results scoring were used new metric proposed by CinC Challenge organizer [5]. Following this paper $C = [c_i]_{i=1}^{m}$ is a collection of m distinct diagnoses for a database of n recordings. $A = [a_{ij}]$ is a multi-class confusion matrix where a_{ij} is the normalised number of recordings in a database that were classified as belonging to class c_i but actually belong to class c_j. Different entries have assigned different weights $W = [w_{ij}]$ defined in [5] based on the similarity of treatments or differences in risks. The score $s = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} a_{ij}$ is a generalized version of the traditional accuracy metric. Finally, this score is normalized so that a classifier that always outputs the true class or classes receives a score of 1 and an inactive classifier that always outputs the normal class receives a score of 0:

$$s_{normalized} = \frac{s - s_{inactive}}{s_{true} - s_{inactive}},$$

where $s_{inactive}$ is the score for the inactive classifier and s_{true} is the score for ground-truth classifier. A classifier that returns only positive outputs will typically receive a negative score, i.e. a lower score than a classifier that returns only negative outputs, which reflects the harm of false alarms.

Results for Scenario A (Tab. 1) are still calculated and will be presented in final version of the article.

Conducting tests for even and uneven branches, we obtained results that were mostly consistent with [8] as can be seen in Tab. 2. Even branches performed better than uneven for GRU and N-BEATS for every experiment. What is curious is that there were configurations for LSTM where the dominant α branch obtained slightly better results (0.379 to 0.377). Still, for most configurations, equal branches obtained

Table 2. Scenario B – comparison between configurations with equal and unequal branch sizes.

Network Configuration	Equal		α dominant		β dominant	
	avg	std	avg	std	avg	std
N-BEATS	**0.32**	0.06	0.27	0.06	0.25	0.04
LSTM	**0.38**	0.01	**0.38**	0.01	0.37	0.01
GRU	**0.39**	0.02	0.34	0.07	0.35	0.01

Table 3. Scenario C – comparison of LSTM, LSTM with peepholes, GRU and N-BEATS for different number of leads. Network configuration: 2 hidden layers of size 7. Results are averages from tests on 5 folds, time is for one peak in milliseconds, avg is average challenge score.

Leads	LSTM			N-BEATS			LSTM peepholes			GRU		
	avg	std	time	avg	std	time	avg	std	time	avg	std	time
12	0.40	0.01	0.0036	0.27	0.07	**0.0002**	**0.42**	0.01	0.0395	0.39	0.03	0.0037
6	0.38	0.01	0.0036	0.22	0.01	**0.0002**	**0.40**	0.01	0.0390	0.39	0.03	0.0036
4	**0.39**	0.01	0.0026	0.27	0.06	**0.0002**	**0.39**	0.01	0.0282	**0.39**	0.03	0.0027
3	0.39	0.01	0.0021	0.30	0.06	**0.0002**	**0.40**	0.01	0.0231	0.39	0.02	0.0022
2	**0.38**	0.01	0.0017	0.35	0.02	**0.0002**	**0.38**	0.01	0.0180	**0.38**	0.01	0.0014

Table 4. Scenario C – Average accuracy and average F_1-score for LSTM, LSTM with peepholes, GRU and N-BEATS models on the training data.

Leads	LSTM		N-BEATS		LSTM peepholes		GRU	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
12	0.020	0.270	0.002	0.116	**0.023**	**0.282**	0.014	0.251
6	0.016	0.243	0.005	0.184	**0.027**	**0.275**	0.020	0.264
4	0.013	0.224	0.002	0.178	**0.016**	**0.260**	0.015	0.252
3	0.011	0.209	0.001	0.160	**0.020**	**0.267**	0.019	0.258
2	0.012	0.237	0.003	0.194	0.014	0.247	**0.014**	**0.249**

the best results.

In Scenario C (test for even $\alpha\beta$ configurations with 2 hidden layers of size 7) best results were obtained by LSTM with peepholes (Tab. 3). GRU and classic LSTM obtained slightly worse results but ten times faster than peepholes. N-BEATS obtained the worst results, but the network started to achieve acceptable results with fewer leads. Interestingly, the speed of the N-BEATS network is a degree of magnitude faster than GRU and LSTM and two degrees of magnitude faster than LSTM with peepholes.

Results for scenario D (Tables 5 and 6) will be presented after finishing the process of fine tuning of the networks in the final version of the article.

Table 5. Scenario D – best results obtained for each neural network architecture. Results are averages from tests on 5 folds, time is in milliseconds, avg is average challenge score. Note: Configurations will be added after finishing the fine tuning process.

Leads	LSTM			N-BEATS			LSTM peepholes			GRU		
	avg	std	time	avg	std	time	avg	std	time	avg	std	time
12												
6												
4												
3												
2												

Table 6. Scenario D – Average accuracy and average F_1-score for LSTM, LSTM with peepholes, GRU and N-BEATS models on the training data.

Leads	LSTM		N-BEATS		LSTM peepholes		GRU	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
12								
6								
4								
3								
2								

3.4. Discussion

Full discussion of the obtained results will be presented in the final version of the paper after all the calculations are complete and validated.

As can be seen in Table 4 both accuracy and F_1-score of tested models are low. It is because to maximize challenge metrics, only diseases belonging to 26 classes that were part of the metric [21] were learned by the networks. Networks that were learned for all existing classes in the dataset were penalized by how the challenge metric was calculated and not presented in the paper.

Remark 1: An interesting observation is that adding the third lead (the chest V2 lead) obtained a lower challenge metric for nine of the ten splits of the training data while obtaining the second-best result for the tenth training split. This interesting behaviour was noted in more then one configuration, additional investigation with help of expert in cardiology is conducted.

Remark 2: There are published official challenge scores for both tested architectures as in the organizer's challenge metric function however they do not reflect real performance of our models; a mismatch in outputs formatting caused an error which ignored float values and returned the lowest possible score. At the moment we are awaiting re-evaluation of our N-BEATS version on hidden data set.

4. Conclusions

In the paper, N-BEATS was compared to other state-of-the-art RNNs adopted to use the $\alpha\beta$ architecture introduced in [8]. In this paper, we extended the experiments conducted in [9] in the field of use of N-BEATS architecture as a multi-label classifier. We modified it into the $\alpha\beta$ scheme and thoroughly compared it with other SotA RNN architectures like LSTM, GRU, LSTM with peepholes. The experiments were carried out on datasets from the CinC challenge using the challenge score as a metric to check the validity of $\alpha\beta$ architecture, effects of dominant branch configurations, speed of each algorithm, and effects of network size and a number of leads. Conducted research focused on the architecture performance and influence of ECG lead reduction for classification results.

Even branch $\alpha\beta$ architecture achieved better results in most cases for all RNNs. The highest challenge metric was obtained by LSTM with peepholes, while the lowest was with N-BEATS. Still, N-BEATS times are two orders of magnitude faster than LSTM with peepholes and an order of magnitude faster than GRU and LSTM. This, combined with acceptable results (especially after some tuning of the network for challenge problem) for a low number of leads, allows for the solution for arrhythmias detection and classification while using off-the-shelf wearable devices (Holter monitors, sport bands, etc.).

Future work includes modifying the presented two-branch architecture of the N-BEATS network with attention blocks, a new powerful tool to enhance the time series analysis.

References

[1] Kiranyaz S, Ince T and Gabbouj M 2015 *IEEE Transactions on Biomedical Engineering* **Vol. 63**
[2] Hannun A Y, Rajpurkar P, Haghpanahi M, Tison G H, Bourn C, Turakhia M P and Ng A Y 2019 *Nature Medicine* **25** 65–69
[3] Jun T, Nguyen H M, Kang D, Kim D, Kim D and Kim Y H 2018 *CoRR* **abs/1804.06812** (*Preprint* 1804.06812) URL http://arxiv.org/abs/1804.06812v1
[4] Ribeiro A H, Ribeiro M H, Paixão G M, Oliveira D M, Gomes P R, Canazart J A, Ferreira M P, Andersson C R, Macfarlane P W, Jr M W and et al 2020 *Nature Communications* **Vol. 11** 1—9
[5] Alday E A P, Gu A, Shah A J, Robichaux C, Wong A K I, Liu C, Liu F, Rad A B, Elola A, Seyedi S, Li Q, Sharma A, Clifford G D and Reyna M A 2021 *Physiological Measurement* **41** 124003 URL https://doi.org/10.1088/1361-6579/abc960
[6] Oreshkin B N, Carpov D, Chapados N and Bengio Y 2019 *CoRR* **abs/1905.10437** (*Preprint* 1905.10437) URL http://arxiv.org/abs/1905.10437v4
[7] Oreshkin B N, Carpov D, Chapados N and Bengio Y 2020 *CoRR* **abs/2002.02887** (*Preprint* 2002.02887) URL https://arxiv.org/abs/2002.02887v3
[8] Saadatnejad S, Oveisi M and Hashemi M 2019 *IEEE Journal of Biomedical and Health Informatics (JBHI)* **2** 515–523
[9] Puszkarski B, Hryniów K and Sarwas G 2021 N-BEATS for heart dysfunction classification *Challanges in Cardiology (CinC) 2021*
[10] Gers F A, Schraudolph N N and Schmidhuber J 2002 *Journal of Machine Learning Research* **3** 115–143
[11] Cho K, van Merrienboer B, Gülçehre Ç, Bougares F, Schwenk H and Bengio Y 2014

Learning phrase representations using RNN encoder-decoder for statistical machine translation *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* pp 1724—-1734

[12] Smyl S 2020 *International Journal of Forecasting* **36** 75–85 ISSN 0169-2070 m4 Competition

[13] Accessed on: 2021-08-30 How the heart works United States Department of Health & Human Services URL `https://www.nhlbi.nih.gov/health-topics/how-heart-works`

[14] Pan J and Tompkins W J 1985 *IEEE Transactions on Biomedical Engineering* **BME-32** 230–236

[15] Liu F, Liu C, Zhao L, Zhang X, Wu X, Xu X, Liu Y, Ma C, Wei S, He Z, Li J and Kwee E N Y 2018 *Journal of Medical Imaging and Health Informatics* **8** 1368—1373

[16] Tihonenko V, Khaustov A, Ivanov S, Rivin A and Yakushenko E 2008 *PhysioBank, PhysioToolkit, and PhysioNet* Doi: `10.13026/C2V88N`

[17] Bousseljot R, Kreiseler D and Schnabel A 1995 *Biomedizinische Technik* **40** 317–318

[18] Wagner P, Strodthoff N, Bousseljot R D, Kreiseler D, Lunze F I, Samek W and Schaeffter T 2020 *Scientific Data* **7** 1–15

[19] Zheng J, Zhang J, Danioko S, Yao H, Guo H and Rakovski C 2020 *Scientific Data* **7** 1–8

[20] Zheng J, Cui H, Struppa D, Zhang J, Yacoub S M, El-Askary H, Chang A, Ehwerhemuepha L, Abudayyeh I, Barrett A, Fu G, Yao H, Li D, Guo H and Rakovski C 2020 *Scientific Data* **10** 1–17

[21] Reyna M A, Sadr N, Perez Alday E A, Gu A, Shah A, Robichaux C, Rad B A, Elola A, Seyedi S, Ansari S, Ghanbari H, Li Q, Sharma A and Clifford G D 2021 *Computing in Cardiology* **48** 1–4