Rhythm Classification of 12-Lead ECGs Using Deep Neural Networks and Class-Activation Maps for Improved Explainability

Sebastian D Goodfellow[1,2], Dmitrii Shubin[1,2], Robert W Greer[1], Sujay Nagaraj[4], Carson McLean[4], Will Dixon[1], Andrew J Goodwin[1, 3], Azadeh Assadi[1], Anusha Jegatheeswaran[1], Peter C Laussen[1], Mjaye Mazwi[1], Danny Eytan[1, 5]

[1] Department of Critical Care Medicine, The Hospital for Sick Children, Toronto, Ontario, Canada
[2]Department of Civil and Mineral Engineering, University of Toronto, Toronto, Ontario, Canada
[3]School of Biomedical Engineering, University of Sydney, Sydney, New South Wales, Australia
[4]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
[5]Department of Medicine, Technion, Haifa, Israel

Abstract

*As part of the PhysioNet/Computing in Cardiology Challenge 2020, we developed a model for multilabel classification of 12-lead electrocardiogram (ECG) data according to specified cardiac abnormalities. Our team, LaussenLabs, developed a novel classifier pipeline with 6 core features (1) the addition of r-peak, p-wave, and t-wave features that were input into the model along with the 12-lead data, (2) data augmentation, (3) competition metric hacking, (4) modified WaveNet architecture, (5) Sigmoid threshold tuning, and (6) model stacking. Our approach received a score of **0.63** using 6-fold cross-validation on the full training data. Unfortunately, our model was unable to run on the test dataset due to time constraints, therefore, our model's final test score is undetermined.*

1. Introduction

Cardiovascular disease is the leading cause of death worldwide [1] and different cardiovascular diseases have different causes and require different interventions, where the electrocardiogram (ECG) is an essential tool for screening and diagnosing cardiac electrical abnormalities [2]. The PhysioNet/Computing in Cardiology Challenge 2020 focused on automated, open-source approaches for classifying cardiac abnormalities from 12-lead ECGs [3, 4]. Our entry for the Challenge applied a novel neural network architecture and training procedures, which are described further in this paper.

2. Methods

The following is an overview of our methodology presented in eight sections (1) Preprocessing, (2) Feature Extraction, (3) Model, (4) Augmentation, (5) Training, (6) Class Activation Maps, (7) Tuning and (8) Inference.

2.1. Preprocessing

ECG waveform training data for this challenge was sampled at 3 different rates (257, 500, and 1000 Hz). Thus, we chose to upsample all ECG data to 1000 Hz using the SciPy **resample** function. Waveform amplitudes were scaled by the median r-peak amplitude on Lead-I if the BioSPPy [5] algorithm was able to successfully pick r-peaks. If the signal was too noisy for reliable r-peak extraction, then each lead was standardized by subtracting its median amplitude and then dividing by the standard deviation of all 12 Leads combined. Our model's input size was 19,000 samples (19 seconds at 1000Hz), which was chosen as the optimal trade-off between training time and model performance on cross-validation. Any samples with a duration less than 19 seconds were zero-padded while samples with a duration greater than 19 seconds were clipped after the first 19 seconds of recorded data.

2.2. Feature Engineering

We extracted three features, which were combined with the ECG signals and input into the model. Features were engineered to indicate the location of r-peaks, p-waves, and t-waves as seen in Figure 1. For each lead, the position of r-peaks, p-waves and t-waves were computed and are visualized as blue, red, and green dots respectively in Figure 1. BioSPPy [5] was used to compute the r-peak locations and our algorithm was used to compute the p-wave and t-waves locations. Our p-waves and t-waves detection algorithm applies a 10 Hz low-pass filter to the R-R intervals and then performs peak finding. R-peak, p-wave

ISSN: 2325-887X DOI: 10.22489/CinC.2020.353

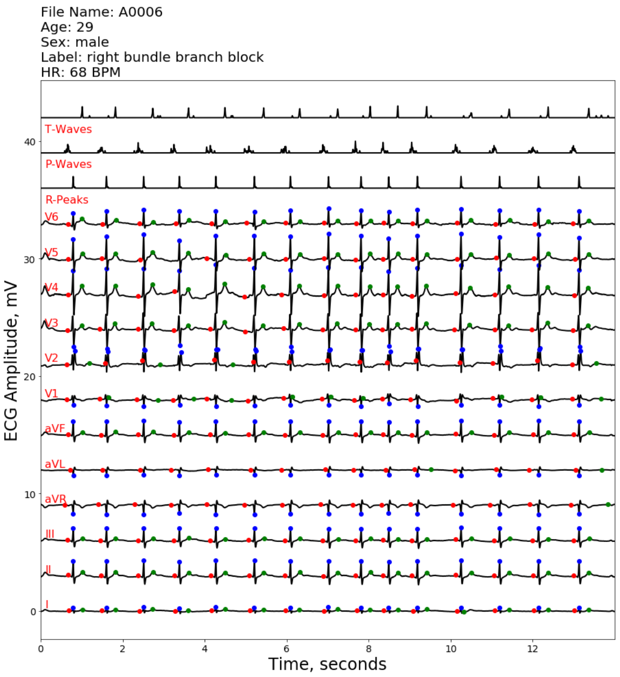


Figure 1. Example of ECG data and r-peak, p-wave and t-wave features.

and t-wave times were combined using a kernel density approach to generate a 1D normalized signal, which effectively provides their likelihood locations. These features were designed to help the model learn these important features thus, helping improve convergence.

2.3. Model

The input to our model is an array containing 12 ECG leads and three engineered features of 19 seconds in duration with a sampling rate of 1000 Hz. The array is configured such that each signal is a separate channel, which resulted in a 15-channel input. The stem consists of two layers each containing a 1D convolution, batchnorm, ReLU activation, max-pooling, and dropout (see Figure 2). The purpose of the stem layers was to downsample to the input signal from 19,000 to 4,750 data points for GPU memory considerations. The output from the stem is input into a series of 8 residual layers (see Figure 2) that are modelled after WaveNet's [6] residual layers with the only difference being that the convolutions are not causal. The 8 skip connections are summed and fed into a series of output convolution layers with the same architecture as the stem layers. The final output is globally averaged in the time dimension and fed into a dense layer with 27 neurons followed by a Sigmoid activation. The outpoint from the Sigmoid function is then squared because the competition metric is less sensitive to false-positive predictions. In contrast,

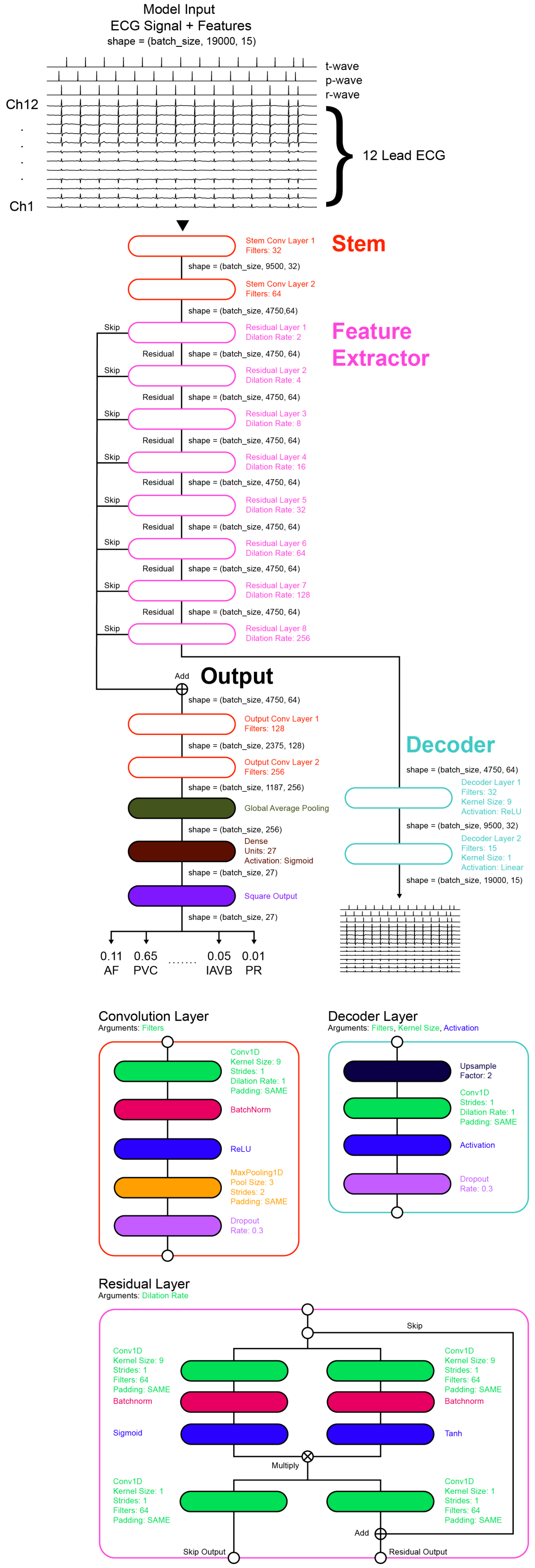


Figure 2. Overview of neural network architecture.

it simulates the uplift prediction over recognizing all signals as normal sinus rhythm. Since Sigmoid outputs are distributed between 0 and 1, the squaring operation forces the model to give a higher penalty to false-negative predictions rather than false-positive predictions. In addition to this output, a decoder was positioned after the last residual layer where two decoder layers upsampled the data to its original input shape (batch-size, 19000, 15). The purpose of this auxiliary output was to improve the feature extraction pipeline. Instead of learning features for classification only, it also tries to represent morphological features of the input ECG signal.

2.4. Augmentation

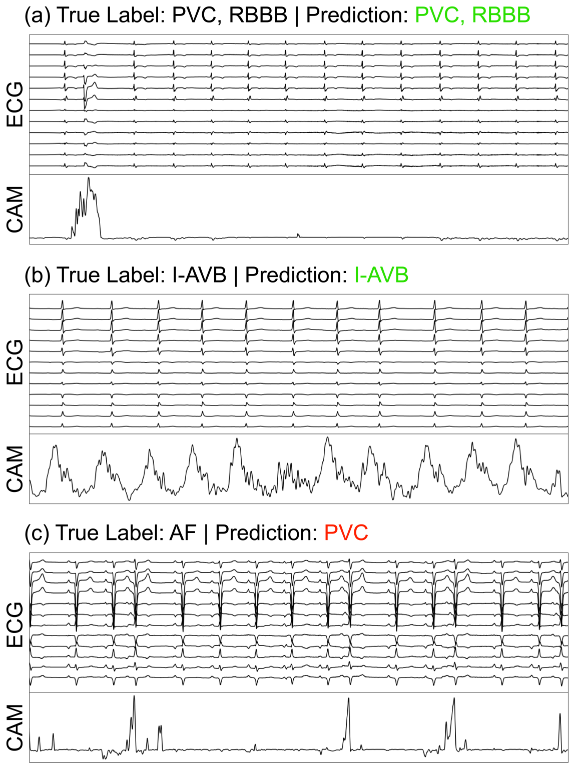
During our exploratory data analysis, we notice many common noise artifacts in the data, such as baseline wandering, and our initial plan was to filter them out during pre-processing. However, when experimenting with the model, we observed superior performance when the data was left unfiltered. Therefore, we decided to augment the data with synthetic noise. We developed four synthetic noise sources that we randomly added to the data during training (1) Gaussian noise, (2) high-frequency/low-amplitude oscillations, (3) baseline wandering, and (4) large-amplitude transient pulses. Two additional augmentation strategies were employed. The first applied a random multiplication factor to the waveform amplitude and the second randomly perturbed the heart rate of the signal. For example, if the true heart rate was 124 BPM, we would add a random fluctuation changing the value to 132 BPM and then resampling to 1000 Hz. This was only performed for training samples where the label was not heart rate dependent.

2.5. Training

For training, the data was split into 6 folds for cross-validation using the open-source package **iterative-stratification**, which is designed for multilabel stratification. The model was trained for 100 epochs with early stopping with and patience set to 10. The learning rate was initially set to 1e-3 (batch size 128) and followed a decay schedule **ReduceLROnPlateau** with patience equal to 1. The loss function was the sum of the binary cross-entropy of the classifier and mean squared error of the decoder and was Optimized using the Adam optimizer [7].

2.6. Class Activation Maps

Our model architecture was designed such that Class Activation Maps (CAMS) could be computed. We followed the 1D CAM formulation of Goodfellow et al. (2018) [8]. Goodfellow et al. (2018) [8] initially conceived of CAMs applied to ECG data for use in explaining arrhythmia predictions to clinicians at the bedside. However, for this competition, CAMs were used to help understand errors made by the model during training. Figure 3 presents three examples of CAMs where (a) and (b) were predicted correctly and (c) was predicted incorrectly. For Figure 3 (a), the model correctly predicts Premature Ventricular Contraction (PVC) and Right Bundle Branch Block (RBBB) and the CAM for PVC clearly shows evaluated activation at the time of the PVC event. For Figure 3 (b), the model correctly predicted First-degree Atrioventricular Block (I-AVB) and the CAM shows elevated activation in the P-R region of each beat, which is consistent with I-AVB's main feature of elongated PR intervals. For Figure 3 (c), the model incorrectly predicts PVC when the true label is Atrial Fibrillation (AF). In this case, the CAM shows four activation spikes coinciding with what look to be PVC events. From here, we engaged our clinical teammates to determine if the training label was correct, for which the CAMs proved useful.


(a) True Label: PVC, RBBB | Prediction: PVC, RBBB

(b) True Label: I-AVB | Prediction: I-AVB

(c) True Label: AF | Prediction: PVC

Figure 3. Example of class activation maps.

2.7. Tuning

Output model predictions are managed by a post-processing pipeline. After training was completed, we applied an algorithm for finding the optimal Sigmoid threshold by iterating over all thresholds between 0.05 and 0.95 with a 0.05 step, calculating the competition metric and select the best one. The optimal threshold was found for the training split and then applied to the validation set. Finding the threshold on the training set prevented leakage and mitigated the influence of incorrect labels.

2.8. Inference

At inference time, the six models trained for each cross-validation split were used for prediction. Hard predictions from each model were combined by a majority vote. This helped improve generalization of the final predictions and mitigated the influence of incorrect labels. Considering that each model was trained on different incorrectly labelled data, the resulting outputs are more independent and therefore, provide a more robust group prediction.

3. Results

Our model's cross-validation scores are presented in Table 1 and show a minimum of 0.614, a maximum of 0.644, and a mean of 0.63. Unfortunately, we were unable to get out model to run on the test dataset by the competition deadline. As a result, we were given a test score of -0.406, which is the score if all predictions are 0 for all test samples.

Table 1. Summary of model cross-validation and test performance.

Dataset	Competition Metric
CV FOLD 1	0.631
CV FOLD 2	0.637
CV FOLD 3	0.644
CV FOLD 4	0.619
CV FOLD 5	0.614
CV FOLD 6	0.640
CV MEAN	**0.630**
CV STDEV	0.010
TEST	**-0.406** **(Submission Error)**

4. Discussion and Conclusions

Our CV score of 0.63 was the result of exhaustive model architecture experimentation and hyper-parameter tuning. Unfortunately, the competition came to a close, however, our next strategy would have been relabelling. We developed a Python application for our clinical teammates to allow them to view ECG samples where the model made an incorrect prediction and provide feedback and label corrections. From inspecting many samples with our clinical teammates, it was clear that a large number of training samples appeared to be miss-labelled. See Figure 3 (c) for a clear example. We also attempted to use gender and age features by concatenating them to the output from the global average pooling layer, however, this resulted in a minor decline in performance and was therefore not implemented.

In conclusion, we develop a novel model architecture and training strategy for the 2020 Physionet/CinC Challenge which produced a CV score of 0.63. Unfortunately, we were unable to get a successful test score by the competition deadline. Our competition code is available at `github.com/Seb-Good/physionet-challenge-2020`.

References

[1] Benjamin EJ, Muntner P, Alonso A, Bittencourt MS, Callaway CW, Carson AP, Chamberlain AM, Chang AR, Cheng S, Das SR, et al. Heart Disease and Stroke Statistics – 2019 Update: a report From the American Heart Association. Circulation 2019;.

[2] Kligfield P. The centennial of the Einthoven electrocardiogram. Journal of Electrocardiology 2002;35(4):123–129.

[3] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–e220.

[4] Perez Alday EA, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, Liu F, Rad BA, Elola A, Seyedi S, Li Q, Sharma A, Clifford GD, Reyna MA. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Under Review 2020;.

[5] Carreiras C, Alves AP, Lourenço A, Canento F, Silva H, Fred A, et al. BioSPPy: Biosignal processing in Python, 2015. URL `https://github.com/PIA-Group/BioSPPy/`. [Online; accessed 2020].

[6] van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. Wavenet: A generative model for raw audio. arXiv preprint arXiv160903499 2016;.

[7] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In Proceedings of 3rd International Conference for Learning Representations, San Diego. 2015; .

[8] Goodfellow SD, Goodwin A, Greer R, Laussen PC, Mazwi M, Eytan D. Towards understanding ECG rhythm classification using convolutional neural networks and attention mappings. In Proceedings of Machine Learning for Healthcare 2018 JMLR WC Track Volume 85, Aug 17–18, 2018, Stanford, California, USA. 2018; .

Address for correspondence:

Sebastian Goodfellow
University of Toronto, Toronto, Ontario, Canada
sebi.goodfellow@utoronto.ca