Detecting Cardiac Abnormalities with Multi-Lead ECG Signals: A Modular Network Approach

Ryan Clark[1], Mohammadreza Heydarian[2], Kashif Siddiqui[1], Sajjad Rashidiani[2], Md Asif Khan[2], Thomas E Doyle[1,2,3]

[1]School of Biomedical Engineering, McMaster University, Hamilton, ON, Canada
[2]Dept. of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada
[3]Vector Institute for Artificial Intelligence, Toronto, ON, Canada

Abstract

Globally, heart disease has been the leading cause of death for more than two decades. There is a need to develop intelligent architectures to handle a variety of real life clinical scenarios when a 12-lead ECG is not a viable option. We propose a method using wide and deep CNN architectures to classify cardiac abnormalities from 12, 6, 4, 3, and 2 leads ECGs. These five networks were created for the PhysioNet/CinC Challenge 2021, by the Biomedic2ai team.

ECG signals were down-sampled to 100Hz and partitioned with 5-second windows using a sliding 4-second overlap. A one-dimensional deep CNN (1D-dCNN) module was used to preserve sequentially related features embedded in the signals. A feature extraction module was added to the 1D-dCNN, creating a 'wide and deep modular network'. This framework allows the addition or removal of modules to optimize classification models.

We achieved test scores of 0.36, 0.30, 0.31, 0.29, and 0.34 (ranked 23rd, 26th, 26th, 27th, and 22nd out of 39 officially ranked teams) for 12, 6, 4, 3, and 2 leads, respectively, on the hidden test set provided by the challenge.

Our model demonstrates potential with the wide modular network. The framework also provides the flexibility to integrate clinical knowledge in the future modules to improve the overall classification performance.

1. Introduction

Cardiovascular diseases (CVDs) are responsible for one-third of the deaths globally [1]. Early diagnosis can be life-saving, with favourable prognosis. Electrocardiogram (ECG) helps screen and diagnose these CVDs by allowing visualization of aberrant conduction in the heart [1, 2]. The 12-lead ECG is a noninvasive cardiovascular diagnostic tool used worldwide. In recent years, developments in deep learning methods have improved the ability of automated systems to detect CVDs from the ECG signals. However, these automated systems lack generalizability due to the absence of sufficient high-quality data, and standardization of performance metrics of classification algorithms [3]. Moreover, the models are often black box systems that seldom explain the decision criterion or provide human interpretable context for their decision, which limits the trustworthiness of such systems.

The PhysioNet/CinC Challenge 2021 focuses on developing automated, open-source methods to classify CVDs from multi-lead ECGs [4–6]. Our entry in the Challenge used a modular approach, combining a 1D-dCNN and a perceptron as individual modules to build a wide and deep model. The model classifies 30 diagnoses using various reduced-lead combinations of the 12-lead ECG signals collected from patients around the world [6].

The rest of the paper is organized as follows: Section 2 provides a detailed description of our proposed model, Section 3 presents the results, and Section 4 discusses the performance and weakness of our model.

2. Methods

The PhysioNet/CinC Challenge 2021 provided six real-world datasets of ECG signals. Due to multi-dimensional and temporally dependent nature of the data, we designed a CNN-based model that has historically been used to classify ECG signals [7]. In addition, a parallel shallow perceptron was employed for heart rate variability (HRV) feature analysis using extracted statistical signal information. In the following subsections, the proposed model and data processing steps are described.

2.1. Feature Extraction

The extracted features for the perceptron were age, gender, time domain heart rate variability (tdHRV), and nonlinear domain heart rate variability (nlHRV) features (see Tables 1 & 2 for details). Both the time-based and the

non-linear HRV features were extracted from lead II because it is most commonly used for rhythm detection and is present in all lead combinations for the challenge [6]. A total of 47 raw features were extracted using neurokit [8]. To reduce high dimensionality and subsequent computation complexity of the extracted features, principal component analysis (PCA) was used to reduce the extracted features from 47 to 10. Tables 1 & 2 depict a sample of the 47 raw features extracted before PCA was conducted. This reduced feature set was then scaled using standardization.

Feature	Explanation
MeanNN	Mean of the RR intervals
SDNN	Std Dev of the RR intervals
SDANN1	Std Dev of avg. RR intervals
SDNNI1	Mean of Std Dev of RR intervals
RMSSD	Sqrt. of mean of sum of successive RR intervals
SDSD	Std Dev of successive diff. between RR intervals
CVNN	SDNN/ MeanNN
CVSD	RMSD/ MeanNN
MedianNN	Median |diff. between RR intervals|
⋮	
HTI	HRV Triangular index

Table 1: Sample of tdHRV features

Feature	Explanation
SD1	Measure of the spread of RR intervals
SD2	Long-term RR interval fluctuations index
SD1SD2	SD1 / SD2
CSI	Cardiac Sympathetic Index
CVI	Cardiac Vagal Index
CSI_mod	Modified CSI
GI	Guzik's Index
SI	Slope Index
⋮	
FuzzyEn	Fuzzy Entropy

Table 2: Sample of nlHRV features

2.2. Signal Preprocessing

The signals were detrended using `signal-detrend` function from `neurokit` library [8] in Python. By detrending, only the differences in values arising from the cyclical nature of the signal and other inherent patterns associated with the CVDs will be identified. A 2nd order butterworth band-pass filter was applied to remove the frequencies lower than 5Hz and higher than 100Hz, using the function `signal-filter` from the `neurokit` library.

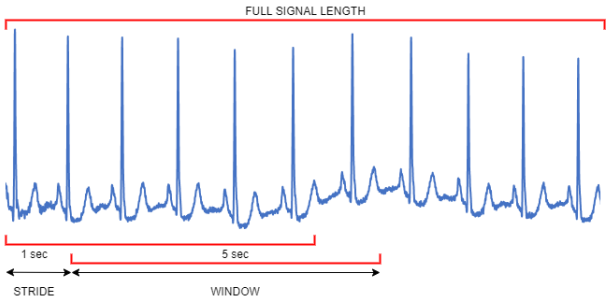

Figure 1: Windowing and stride of a filtered signal

The six different datasets available for this challenge [5, 6] had three different sampling frequencies: 1000Hz, 500Hz, and 257Hz. Fourier transform was used to downsample all signals to 100Hz to provide the classifier with consistent time-scale data and to reduce the size of the overall dataset.

The signals were then segmented into 5-second windows with a stride of one second (see Figure 1). This resulted in 4-second overlap for the adjacent signals (e.g., a 10-second signal will be transformed into six 5-second signals).

A standardization method was applied on the windowed signal and on the selected extracted features separately and both scalars were used to modify the testing data as well.

2.3. Wide and Deep Model

The wide and deep model architecture, visualized in Figure 2, allows the model to combine ECG extracted features and statistical patterns found within the signals themselves, to produce a more robust multi-label classification. Five different models were created to classify the recordings. Each model retains the core architecture, except the input layer dimensions were changed in each model to accommodate the different lead combinations.

The models consist of two modules: i) a deep CNN for converting the raw signal into a series of embeddings by learning the patterns within the recording, and ii) a shallow perceptron network for finding patterns in the extracted features. The deep network consists of four convolution blocks. After the convolution blocks, a flatten layer is used to convert the 2D shape of the patterns to a vector of embeddings. Two fully connected layers are used to reduce the dimension of the embedding vector. The modular nature of the model allows adding a deep CNN network and a shallow perceptron to create an overall 'wide' model.

2.3.1. Deep Module

The deep module of the network consists of four convolution blocks to capture sequentially related features which

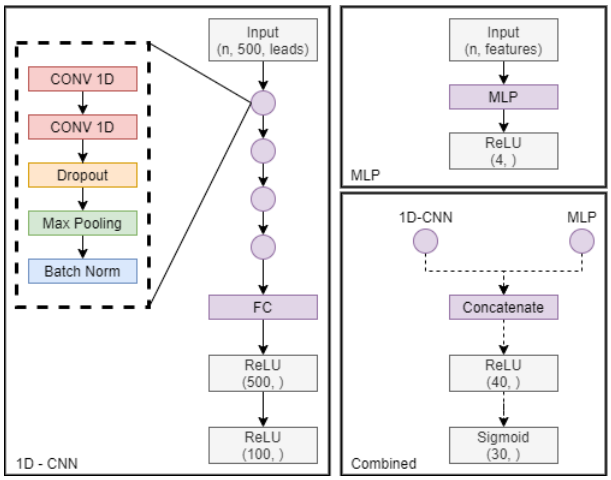

Figure 2: Visualization of the wide & deep model

are embedded in the ECG signals. Each CNN block includes two 1D convolution layers, a dropout layer, a max pooling layer, and a batch normalization layer; see Figure 2 for a visualization of the model's organization. The number of filters in each convolution layer is the same for each layer in the block. The number of filters for the block increases by a factor of two for every convolution block (i.e., 64 for block one, 128 for block two, and 256 for blocks three and four). Each convolution layer uses a filter size of (3x1). To reduce the complexity of the convoluted signals, a max pooling layer is used with a mask size of two. This halves the length of each signal, but keeps the most important activations. To help prevent overfitting, a dropout layer with the rate of 0.3 was included after the second convolution layer in the block. At the end of each block, a batch normalization layer is used to standardize the processed outputs of the convolution block for use in the next block. In the deep module of the network, the preprocessed signal (see section 2.2) is passed through the convolution blocks, then flattened. To reduce the dimension and finding non-linear patterns, two dense layers of 500 and 100 neurons were added after the flatten layer.

2.3.2. Wide Module

The wide module of the network is a neural network (NN) with two layers. The first layer is an input layer of 10 neurons, which takes the 10 extracted features by PCA from the preprocessing step. The second layer has four neurons to further reduce the dimensionality of the features and to identify non-linear patterns between the features.

The outputs of the perceptron and the deep modules are concatenated to form a layer of 104 neurons: 100 are from the deep module output, and 4 are from the perceptron module output. This layer is then passed to a layer with 40 neurons and then to the final output layer, consisting of 30 neurons with a sigmoid activation function; one neuron for each scored classification.

2.4. Model Training and Testing

Binary cross-entropy was used for loss function calculation and training, which is commonly used for multi-label classification problems. The Adam optimizer was used for finding parameters resulting in minimal loss, with a learning rate of 0.0001. The models were trained for a maximum of 20 epochs with the batch size of 32, however early stopping was used as a method of preventing overfitting of the model. Hence, the number of epochs the models were trained on rarely completed the 20 epoch limit. To accomplish this, the training data were split into training set and validation set where the training data consist of 85% of the total training data. The model was trained until the validation loss stopped decreasing for three sequential epochs (i.e., patience).

Training the model was done on the preprocessed signal lengths of five seconds. However, both the training and the test datasets contained signals with signal lengths greater than 5-second. As there is no annotation for each heartbeat of the signal, selecting a 5-second of the signal that has some information of abnormalities is a significant challenge. To overcome this challenge, we selected all 5-second windows of the signal starting from time 0 using the stride of 1 second to produce n overlapping windowed signals. All related overlapping n windowed signals are passed to the classifier sequentially, and any classes predicted for an individual window will be applied to the final prediction of the overall signal.

The augmentation process was implemented by calculating the maximum probability of each class that resulted from applying the classifier on all 5-second segmented signals. This method was chosen to ensure that less common conditions could still be scored positive, even if that condition only appeared in the small segment of the entire signal.

For local evaluation, 85% of the public data were saved as training and validation data, and 15% were saved as testing data. After that, the model was trained and validated on the training data, and the test data was used to evaluate the model using the testing code provided by the challenge organizers. The output of testing was then used to score the model using the challenge metric, also provided by the challenge organizers.

3. Results

Table 3 shows the evaluated challenge scores for our best entry on the public offline training set along with the scoring on the hidden validation set, and the final scoring

on the hidden test set. It also has the ranking of our model on the hidden test set among the 39 ranked teams out of the 68 teams in the challenge.

The model scored the highest for the 12-leads obtaining challenge score of 0.52 on the validation data and 0.36 on the hidden test data.

Leads	Training	Validation	Test	Ranking
12	0.65	0.52	0.36	23
6	0.33	0.39	0.30	26
4	0.64	0.42	0.31	26
3	0.62	0.40	0.29	27
2	0.64	0.41	0.34	22

Table 3: Best challenge scores among our (team Biomedic2ai) several trials on the public training set, best entries on the hidden validation set, and one-time scoring on the hidden test set along with the ranking on the hidden test set.

4. Discussion and Conclusions

There are two important parts in the process of developing a classifier, preprocessing of the data and designing the model. For designing the model, we designed a wide and deep model to be able to use the raw data from the ECG signals in addition to the lead II statistical data which was a common lead in all lead montage combinations. For both modules of the model, hyperparameters were tuned, the number of CNN layers were explored, and the size and the number of filters were refined for enhanced performance. For the preprocessing component of this work, we chose the parameters to reduce memory and the computation resources.

Developing the wide model by combining the perceptron in parallel with the dCNN significantly improved the performance of the classifier with respect to earlier models using only dCNN. Using the wide model at the second (official) phase of the challenge overcame the low performance of the classifier from the first (unofficial) phase of the challenge for the 6-lead combination compared to the rest of the lead-combination scores.

our score for classification of the 6-lead data is lower than that of 4-lead, 3-lead and 2-lead. Our assumption is that more leads, and more data, should result in higher performance; however, there may be redundant information in the 6-lead signals that confused the model. we note this as a limitation of our approach that requires further study. One way to solve this unexpected outcome might be the use of different or customized modules for each combination of the leads.

Similar to other medical AI challenges, highly imbalanced data presents a significant challenge that results in poor predictive performance, especially for the minority classes. As an additional consideration for future study, we intend to employ and develop class balancing techniques to overcome this issue for real-time medical data.

Acknowledgments

The authors would like to acknowledge the support of the Canadian Department of National Defence IDEaS under award number CFPMN2-17.

References

[1] Cardiovascular diseases (CVDs). https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Accessed: 2021-08-17.

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

[3] Strodthoff N, Wagner P, Schaeffter T, Samek W. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. IEEE Journal of Biomedical and Health Informatics 2021;25(5):1519–1528.

[4] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 2000;101(23):e215–e220.

[5] Perez Alday EA, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, et al. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Physiological Measurement 2020;41.

[6] Reyna MA, Sadr N, Perez Alday EA, Gu A, Shah A, Robichaux C, et al. Will Two Do? Varying Dimensions in Electrocardiography: the PhysioNet/Computing in Cardiology Challenge 2021. Computing in Cardiology 2021;48:1–4.

[7] Li J, Si Y, Xu T, Jiang S. Deep Convolutional Neural Network Based ECG Classification System Using Information Fusion and One-Hot Encoding Techniques. Mathematical Problems in Engineering Dec 2018;2018. ISSN 1024-123X.

[8] Makowski D, Pham T, Lau ZJ, Brammer JC, Lespinasse F, Pham H, et al. NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing. Behavior Research Methods Feb 2021;53:1689–1696. ISSN 1554-3528.

Address for correspondence:

Dr. Thomas E Doyle
1280 Main St W, Hamilton, ON L8S 4L8, Canada
doylet@mcmaster.ca