ECG Cardiovascular Disease Diagnosis using Machine Learning

(Convolutional Neural Network/Random Forest)

Abraham Paroya
Department of Electrical and Computer Engineering, Temple University
paroyaabraham@temple.edu

Introduction: In this final exam, the aim was to try two machine learning techniques, one non-neural and one neural to detect six distinct cardiovascular diseases (1dAVB, RBBB, LBBB, SB, ST, and AF) from a dataset comprising 8-channel electrocardiograms (ECG) signals. The dataset was provided in a standardized format, consisting of a data file (.dat) and a header file (.hea). Each signal maintained a consistent sampling rate of 300 Hz, with 2200 samples per record, resulting to approximately 7.33 seconds of ECG recording per record. Before undergoing analysis through two machine learning methodologies, the signals underwent preprocessing utilizing the wfdb library. This preprocessing step involved converting the data into a 3D NumPy array, structured as (total number of records, number of samples per channel, number of channels). This data was then fed into each method, one record at a time, for training. Sklearn, keras and tensorflow were used as the main backbone in developing the two approaches used.

Convolutional Neural Network (CNN): The approach used presents an architecture tailored for detecting cardiovascular diseases in electrocardiograms (ECGs), leveraging both Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) layers. CNNs are particularly well-suited for processing ECG signals due to their inherent ability to capture spatial features in data. ECG signals, though sequential in nature, contain vital spatial information reflecting the electrical activity of the heart. By employing Conv1D layers with specific filter sizes and kernel configurations, the CNN component of the model adeptly extracts hierarchical features from the ECG signals, discerning patterns indicative of various cardiovascular conditions. The subsequent MaxPooling1D layers down sample the extracted features while preserving their characteristics. Additionally, the incorporation of LSTM layers was used to analyze temporal dependencies within the ECG signals. Unlike traditional feedforward architectures, LSTM networks create a new layer of nodes that map out temporal patterns in ECGs. Therefore, combining the spatial feature extraction capabilities of CNNs with the temporal pattern recognition capabilities of LSTMs, hopes to improve precision, recall and accuracy in predicting these diseases. Table 1 shows the total number of parameters used for this model. Adam was used for the optimization portion of this model with binary cross entropy loss function and a learning rate of 0.001. The final dense layer used a sigmoid activation function to make the final predictions. Many of the individual parameters were chosen arbitrarily and remained constant during testing to further study the effects an LSTM layer has on making predictions. The model was trained using 12 epochs.

Layers	**Parameter Count**
Conv1D	192
MaxPooling1D	0
Conv1D	6206
MaxPooling1D	0
LSTM	33024
Flatten	0
Dense	8320
Dropout	0
Dense	774
Total Parameters:	48516

Table 1 - RNF Macro F1 Scores

Random Forest (RNF): Random Forest (RNF) classifiers are an ideal choice for detecting cardiovascular diseases due to their robustness against overfitting, suitability for high-dimensional data, and ability to handle imbalanced datasets. For this approach, six RNF classifiers are trained, each specializing in identifying different cardiovascular conditions. These classifiers learn from ECG signal features extracted from training data and predict the presence of specific abnormalities in testing data. RBF's ensemble-based approach aggregates predictions from multiple decision trees, ensuring both accuracy and efficiency in diagnosing each cardiovascular condition. For this method, the only parameter was the number of estimators of each RNF which was set to 100.

Results:

Disease	/train - F1 Score	/dev - F1 Scores
1dAVB	0.8796	0.0286
RBBB	0.9773	0.8746
LBBB	0.9600	0.8258
SB	0.9453	0.6246
ST	0.9329	0.0473
AF	0.8796	0.0286

Table 2 - RNF Macro F1 Scores

Disease	/train - F1 Score	/dev - F1 Scores
1dAVB	0.9710	0.7606
RBBB	0.9857	0.9372
LBBB	0.9796	0.9020
SB	0.9782	0.8532
ST	0.9729	0.8620
AF	0.9710	0.7606

Table 3 - CNN Macro F1 Scores

The F1 scores obtained from closed loop testing during training for the RNF exhibited good performance across all categories. However, upon evaluation on the dev set, there was a noticeable decline in performance for diseases 1dAVB, ST, and AF. This decline, especially evident in AF, could potentially stem from its frequent co-occurrence with other cardiovascular diseases. Surprisingly, the RNF model trained on both the train and dev sets showed unexpectedly low F1 scores in these categories. On the other hand, the CNN model demonstrated excellent performance across all categories during training, with all F1 scores surpassing 0.97. Nevertheless, in the dev set, both 1dAVB and AF still exhibited underperformance, each yielding an F1 score of 0.7606. It is intriguing to note that these two diseases exhibited identical F1 scores, whereas other categories displayed similar yet distinct values.

Conclusions:

In summary, both the CNN and RNF methods showcased decent performance in detecting cardiovascular diseases from ECG signals. The CNN model required approximately 7 hours for training, leveraging an NVIDIA A40 GPU on nedc_012. Conversely, the RNF method took about 12 hours for training on nedc_130, with an aim to reduce file I/O time. To further refine the CNN method, I would have explored variations in kernel sizes and increased the number of training epochs. For the RNF, experimenting with diverse data preprocessing techniques, such as transforming data into the frequency domain, could have been beneficial. Additionally, reducing the proportion of healthy exams in the training dataset might enhance model performance. The 50/50 split between healthy and unhealthy cases could potentially lead to overtraining on healthy samples, undermining the model's ability to accurately predict unhealthy cases.

	Data Set		
Algorithm	**Train**	**Dev Test**	**Eval**
CNN	99.52%	97.03%	97.15%
RNF	98.72%	93.29%	93.25%

Table 4. Macro Accuracies