Comparison of a Convolutional Neural Network and a Random Forest Classifier on ECG Data

Anna Chau
Department of Electrical and Computer Engineering, Temple University
anna.chau@temple.edu

Introduction: When it comes to machine learning algorithms, there are numerous approaches that can be taken. Some approaches are utilized in different domains and have different strengths. Some algorithms are more straightforward and easily understood/visualized, but others are much more complex and require more attention to fully understand its concept. This paper will focus on the comparison of a non-neural network versus a neural network when detecting cardiovascular abnormalities within electrocardiogram (ECG) data. Specifically, the following abnormalities will be highlighted in this paper: 1st Degree AV Block (1dAVb), Right Bundle Branch Block (RBBB), Left Bundle Branch Block (LBBB), Sinus Bradycardia (SB), Atrial Fibrillation (AF), and Sinus Tachycardia (ST). More specifically, a Convolutional Neural Network (CNN) and a Random Forest classifier (RNF) will be chosen for this algorithm comparison. Through analyzing metrics such as F1 scores and confusion matrices, we can get an understanding on how these algorithms perform on medical signal data and how accurate they are in detecting abnormalities.

Algorithm No. 1 Description: The Random Forest classifier (RNF) is a common machine learning algorithm known for its effectiveness in classification tasks. It operates by creating numerous decision trees during its training process, with each tree in the forest being built using a random subset of samples or features of the training data. During evaluation, the input evaluation data is processed through each decision tree, and the final classification of the algorithm is determined through a majority vote of all the individual decision trees, or an averaging of these decision trees. Because we are focused on diagnosing medical signal data, RNF is known to provide insight on feature importance when splitting decision tree nodes, which may be essential when understanding which features of the ECG are most important to each of the six classes.

Because it is possible for one ECG exam to have multiple cardiac abnormalities, it would be more beneficial to train six different random forests on each of the six cardiovascular abnormalities. That way, each model/forest can focus on learning the specific features of its target class, which would likely lead to better performance for each individual class. Utilizing the train dataset provided by Dr. Picone, these six random forests can be trained on this dataset with the corresponding annotations that would be isolated for each of the six forests. By training six individual forests with one hundred decision trees *(n_estimators = 100)*, we will be able to utilize these forest models to eventually predict on the evaluation dataset.

Algorithm No. 2 Description: A Convolutional Neural Network (CNN) is one of the many deep learning algorithms that is designed for processing images and time-series data. Utilizing convolutional layers, the CNN architecture extracts features from its input data and applies filters or kernels to this input. It then is typically followed by pooling layers, which down samples the feature maps to extract key features of the input. As the key features are extracted, the weights are continuously re-calculated to prevent vanishing gradients. As these connections between layers are created and strengthened, the CNN can learn the representations of features and complex patterns to ultimately classify the input data. Because CNNs can implement feature engineering, they are commonly used in medical imaging applications.

In my CNN architecture, there are (3) 1D convolutional layers, (2) max-pooling layers, (1) flatten layer, and (2) dense layers. As the CNN takes in the input signal of training data, it applies multiple convolutional filters utilizing the 'relu' activation function. Specifically, the Convolutional Activation Layer consists of 32 filters with a kernel size of 3 and a stride of 1. To further reduce the dimensionality of the feature map, max pooling is performed using a 2x2 max-pooling window in the MaxPooling Layer. This process is repeated with additional convolutional layers and max-pooling layers. The Flatten Layer then transforms the feature map into a flat array. The first Dense Layer, with 64 units, applies the ReLU activation function to learn various sets of weights and biases. Following this, the second Dense Layer, comprising 6 units, activates with the sigmoid activation function to produce probabilities for each class independently. The model is compiled with the Adam optimizer for weight updating, optimizing the binary-cross entropy loss function, which measures loss between the predicted probabilities and true labels during training.

Results:

	Data Set		
Algorithm	**Train**	**Dev**	**Eval**
RNF	0.6728	0.3932	0.4043
CNN	0.9333	0.6482	0.6458

Table 1. F1 Macro Score Comparison of RNF versus CNN across multiple datasets

Algorithm	Data	1dAVb	RBBB	LBBB	SB	AF	ST
RNF	**/train**	0.9992	0.9996	0.9994	0.9993	0.0197	0.0198
RNF	**/dev**	0.0296	0.8759	0.8209	0.6250	0.0046	0.0021
CNN	**/train**	0.9919	0.9981	0.9956	0.9948	0.9931	0.9919
CNN	**/dev**	0.3124	0.9134	0.8610	0.8015	0.6890	0.3119

Table 2: F1 Score Comparison for Individual Diseases

Conclusion:

When analyzing these tables, we can see that the Convolutional Neural Network (CNN) performed better than the Random Forest Classifier (RNF). In specific areas where RNF performed poorly, CNN performed consistently better. This may be because CNNs can automatically learn hierarchical representations of the data. As this network is trained, it can extract features from the raw input data, allowing for complex features to be understood and comprehendible. More specifically, CNNs can leverage the convolutional layers to effectively learn the temporal dependencies in these electrocardiograms, which allows them to directly learn the feature from the raw data. Considering these characteristics of CNNs, this explains the consistently high F1 scores for CNN's training predictions. On the other hand, the CNN did struggle with the development predictions. The two classes, *1dAVB and ST*, were the two lowest performances of the CNN on the dev dataset, which may be because these two cardiac abnormalities have similar signal features.

In terms of the Random Forest Classifier, this ensemble classifier may have limited capacity as it could struggle to learn these features effectively. Because there is no feature extraction layer like a CNN has, the RNF may simply just be creating decision trees based on the input sample and its corresponding annotation, but not delving into the data and its features directly in a way that complex features can be understood. To mimic the behavior of a CNN, manual feature extraction and declaration could be included in a RNF to potentially improve performance. Along with that, RNF models generally struggle to generalize well to unseen data. Therefore, this could explain why in *Table 1*, the macro F1 score for RNF's train and dev decreases. If the train and dev data were combined and trained on for a model, it is likely that the RNF would be able to perform better on this newly trained dataset. Lastly, RNF is also unable to do multi-class classification, which further explains why RNF had a lower performance than CNN.