ECG-Based Abnormal Heartbeat Classification:
A Deep Learning Approach for Arrhythmia Detection

Introduction

Electrocardiograms (ECG) have created a profound impact in the field of cardiology, specifically in recognizing heart arrhythmias, a problem with the rhythm of one's heartbeat. Noninvasive arrhythmia analysis is based on multiple electrodes that reflect the electrical activity on ECGs. An estimated three million cases of arrhythmia occur in the United States yearly (Mayo Clinic). Diagnosing this disease early is the key to one's wellness, yet 18% of ECGs containing Atrial Fibrillation are misinterpreted by cardiologists (Anh et al, 2006). With the recent advancements in technology, Machine Learning algorithms such as Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), allow a model to learn features and identify patterns within a given dataset. Hence, making it possible to autonomously recognize diseases in ECGs, capable of identifying arrhythmias to the accuracy of Cardiologists.

Objectives/Hypothesis

Question: Is it possible to create a model capable of surpassing the accuracy of Cardiologists in identifying heart arrhythmias in Electrocardiograms?
Hypothesis: It is possible to exceed the accuracy of Cardiologists when compared to that of a Convolutional Neural Network's, to identify heart arrhythmias in Electrocardiograms.
Constants: The raw data obtained from the PhysioNet database.
Manipulated Variables: The number of layers in each constructed Convolution Neural Network, the number of hyperparameters within each layer, and the level data augmentation the data extracted from the dataset goes through.
Responding Variables: The loss of the model, measured using the Cross-Entropy Loss function, and accuracy of the model in identifying the presence of an arrhythmia.

Background: Deep Learning

- Deep Learning is a subclass of Machine Learning, which is inspired by a neuron's structure, and function in the brain, groups of neurons are called Neural Networks.
- The first layer of a Neural Network is called the input layer and is composed of neurons that represent the input data.
- A neuron holds a number (often between 0-1), the number corresponds to the activation of each neuron.
- The layers in the middle are hidden layers. These layers contain neurons that are responsible for identifying features within the dataset.
- The activations in neurons of each layer change to correctly predict the right class.
- The last layer of a Neural Network is called the output layer, which contains a neuron for each class in the dataset. Each neuron's activation signifies the model's certainty for that class. Hence, the largest activation in the output layer resembles the model's most confident output.

Procedure: Database

The PhysioNet database contains 8,522 ECG recordings, divided into four classes: Normal, Atrial Fibrillation, Other, and Noisy. The raw data was provided in EFDB-compliant MATLAB V4 files, which including a .mat file containing the ECG recording and a .hea file containing the metadata for the recording (Clifford, et al, 2017).

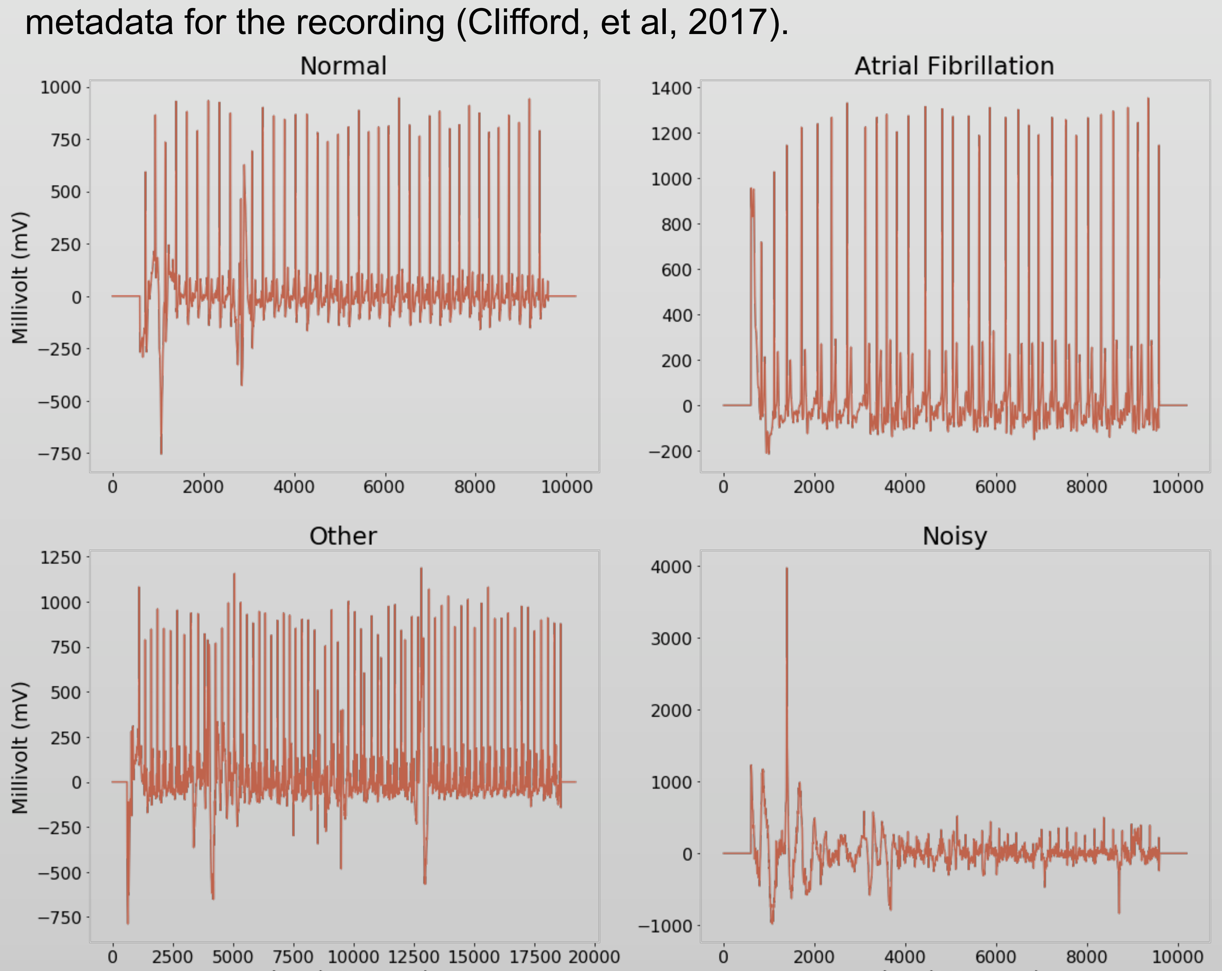

Figure 1: Example data from dataset for each class

Procedure: Pre-Processing Data

- Neural Networks require a constant input vector length. Hence, the raw ECG data was split into sequences, each with a length of 600 indices.
- These slices were based on each peak in an ECG. The peak is considered the middle of the sequence, and a margin of 300 indices on each side of the peak creates a full sequence.
- Each ECG sequence was normalized to values between 0 and 1 to create uniformity in the dataset.
- To create an unbiased model, all classes (e.g. Noisy) in the training data should contain an equal amount of sequences. Thus, the dataset was dramatically reduced to create a fully balanced distribution.

Procedure: Data Augmentation

- A strategy that enables a significant increase in the diversity of data available for training models, without actually collecting new data.
- **Zero Padding**: Appends zeros to the end of an ECG sequence that is not 600 indices in length.
- **Random Zero Bursts**: Implements random zeros in ECG sequences to replicate and constitute noisy data that occurs while collecting a sample
- **Random Resampling**: Changes the sampling rate of an ECG sequence, which stretches or compresses the sequence

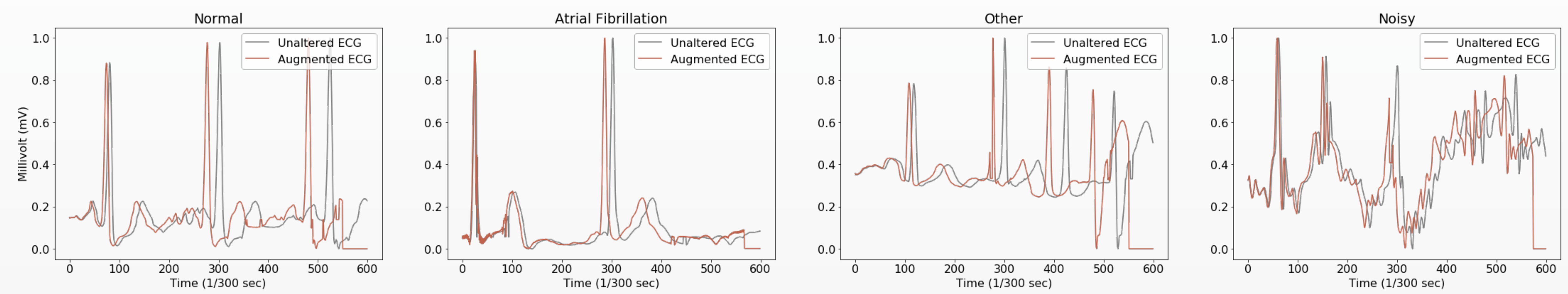

Figure 2: Example of augmented ECG for each class

Procedure: Convolutional Neural Network Model

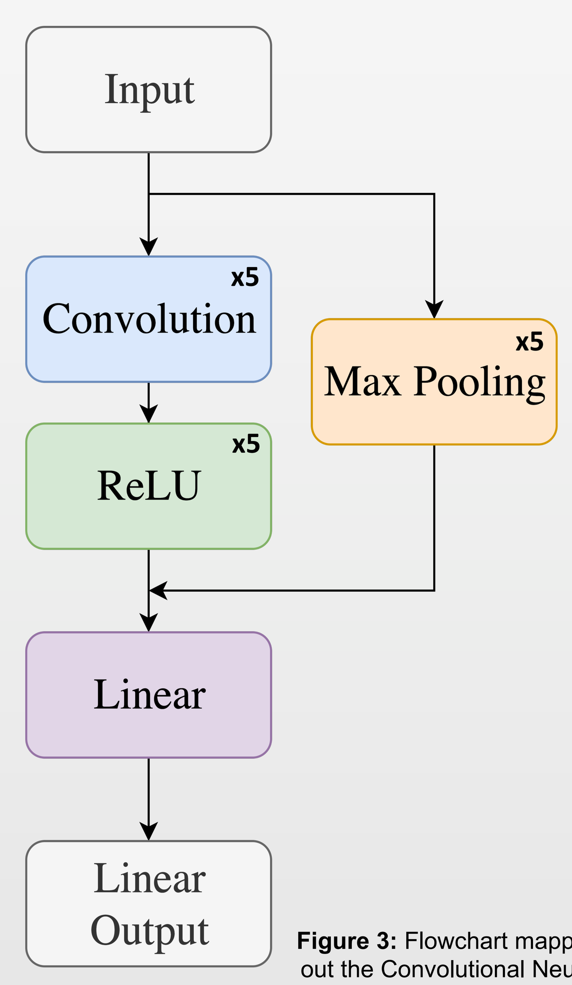

Figure 3: Flowchart mapping out the Convolutional Neural Network

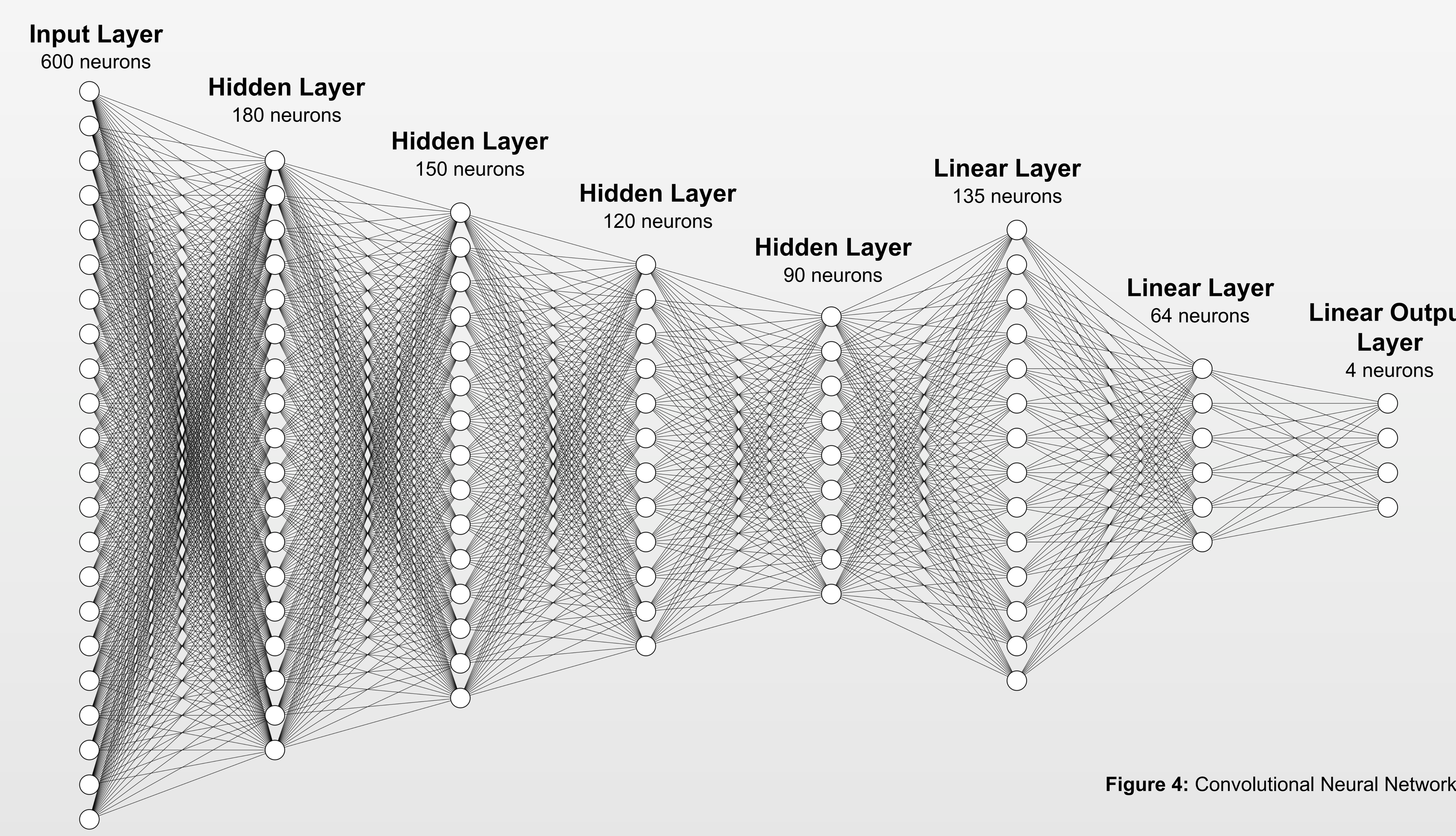

Figure 4: Convolutional Neural Network

- The **input** consists of a 1-dimensional tensor (1x1x600) with a length of 600. The elements are normalized to values between 0 and 1.
- **Convoluting** also decreases the size of the input vector. A vector (with a size of 1x5) filters across the data by producing all values in the vector by a filter vector. This method also assists the model in finding features.
- The **Rectified Linear** activation function alters the range of the incoming data by setting all numbers below 0 to 0 and leaving all positive numbers intact.
- **Max Pooling** reduces the size of the input, in the case of a 1D tensor. A kernel block (with a size of 1x3) filters over the 1D tensor. Taking the maximum value in the kernel block. This operation aids the model in finding features in the data.
- The **Liner** function flattens the incoming result (1x135) into a 1D tensor (1x64).
- The **Linear Output** layer transforms the Linear layer output into a 1x4. Each column in the tensor represents a class's likelihood of being the correct class in the dataset. Thus, the column with the largest values is the model prediction for the input.

Procedure: Evaluation Metrics

Cross-Entropy Loss:

- Measures how good a prediction from the CNN does in terms of being able to predict the expected outcome.
- Aids the CNN in adjusting the weights and bias of a model

Equation:

$$\mathcal{L}(x, c) = -\ln \frac{e^{x[c]}}{\sum_{i=1}^{N} e^{x[i]}}$$

where…
N = Number of classes
c = Index of the correct class
x = Vector of predicted probability of classes

Accuracy:

- Measures the correctness of the CNN's prediction with the ground truth (provided annotation)

Equation:

$$A_n = \frac{2Nn}{\sum_N + \sum_N}$$

$$A = \frac{\sum_{n=1}^{i} A_n}{i}$$

where…
N = Predicted classification for each class
n = Reference classification for each class

Training Results: Loss and Accuracy

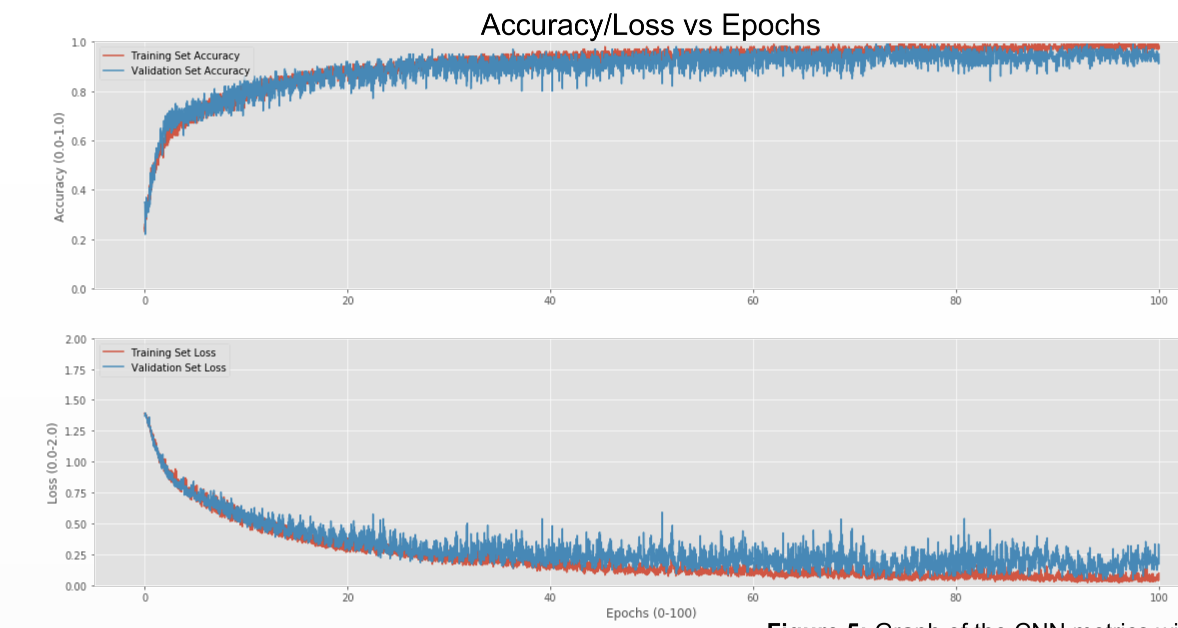

Figure 5: Graph of the CNN metrics with data augmentation

The model was trained on data that was augmented to a factor of 8 and had a total of 653 hyperparameters. The graph above illustrates that the CNN's training and validation set accuracy correlated to each other, implying the CNN is learning, rather than memorizing the training data. Furthermore, both accuracy curves approach 99%. Although the validation loss curve is more sporadic, both loss curves tread similarly, and approach a loss of 0.1.

Training Analysis: Confusion Matrix

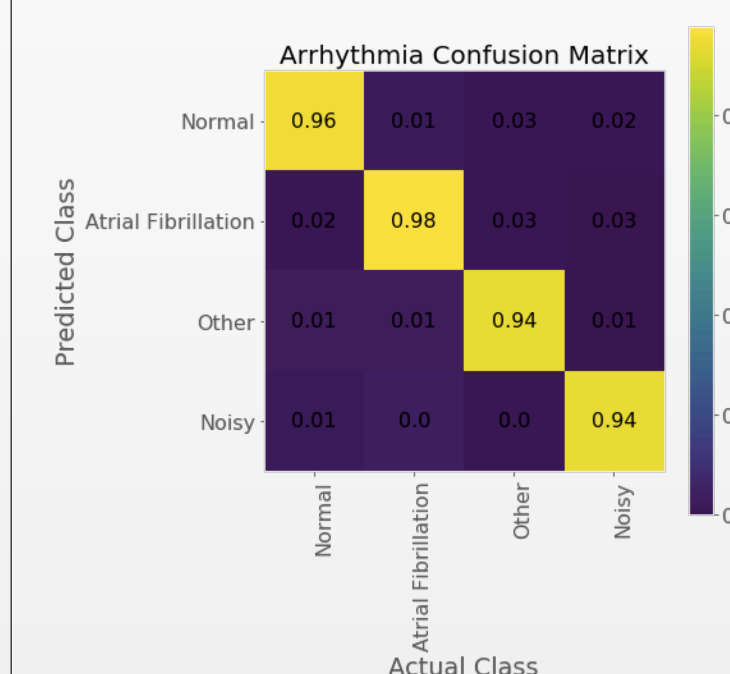

Figure 6: Confusion matrix on model predictions for each class in the training data

The confusion matrix conveys the mistakes the model makes for each class. The most common misunderstanding occurs in the class of other arrhythmias, confusing it with Atrial Fibrillation or a normal ECG, 3% of the time. Largely because the other class consists of many arrhythmias which generally are not structurally alike. Thus, it makes it difficult for the CNN to account for all features of arrhythmias. However, this issue can be solved by replacing the other class with specific arrhythmia classes.

Conclusions

Heart arrhythmias are irregular rhythms in heartbeats that affect 3 million people worldwide every year. Due to the increasing rate of ECGs recording for diagnosis, it is now possible to devolve a Convolutional Neural Network to identify arrhythmias in ECGs. A CNN was developed and trained, to achieve high accuracy in identifying arrhythmias in ECGs. The 1D Convolution Neural Network not only surpassed the accuracy of cardiologists in identifying Atrial Fibrillation, but also achieved an overall top accuracy of 99%, and a constant accuracy of 96%. Furthermore, the CNN was cross-validated against a new dataset that the model had never seen before to ensure no overfitting occurred during the training process. On this test, the CNN model achieved an accuracy of 96%. The key to achieving such success is due to the large annotated dataset (PhysioNet), and data augmentation techniques. Originally, training a shallow CNN with few parameters were thought to create less complexly in learning, and make the CNN faster in training. Doing that merely did the opposite, as the CNN started to overfit to the training data. Adding data augmentation not only fixed the issue of overfitting, but also increased the dataset size; conversely, this increased the time the CNN took to train.

Further Exploration and Application

- Implementing larger datasets with multiple nodes that record the heart's electrical activity simultaneous
  - Apnea-ECG Database
  - CTU-UHB Intrapartum Cardiotocography Database
  - Fantasia Database
  - MIT-BIH Polysomnographic Database
  - OB-1 Database
- Optimizing the time taken to identify an arrhythmia ECG
  - Allows for faster training and response times
- Apply the CNN to an Electroencephalogram (EEG), which measures neural electrical activity to predict body movement, and thought.
- Discovering new methods in identifying arrhythmias in ECGs
- Implement model in ECG reader to autonomously identify arrhythmias in emergencies
- Decrease the number of misdiagnosis in arrhythmias