A Deep Learning Approach for Arrhythmia Detection

***Abstract*—Early detection of cardiac arrhythmia has the potential to prevent the millions of moralities that the disease causes globally. However, there are few automated systems to identify arrhythmia. A significant impediment in achieving successful methods include the lack of a large training dataset. Despite this difficulty, processes like data augmentation allow for an increased amount and diversity of data. Here, the electrocardiogram (ECG) datasets were obtained from the PhysioNet database. The dataset was used to train a Convolutional Neural Network (CNN) on classifying cardiac arrhythmia. Experimental results illustrate advantages such as better responsiveness and higher accuracy of deep learning-based models when compared to the traditional analysis on ECGs.**

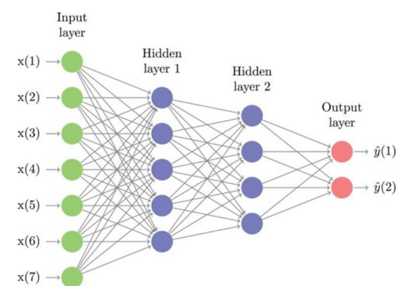

Fig. 1. Diagram of Neural Network

I. 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. Non-invasive 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 mathematical 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.

II. 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.

III. Methods

Database

The PhysioNet database contains 8,522 ECG recordings, divided into 4 classes: Normal, Atrial Fibrillation, Other, and Noisy. The raw data is 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).

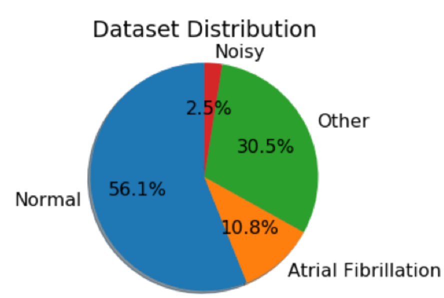

Fig. 2. PhysioNet dataset distribution of classes (Normal, Atrial Fibrillation, Noisy, and Other).

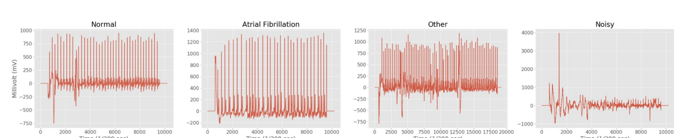

Fig. 3. Example of raw ECG recording for each class, extracted from the PhysioNet dataset

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.

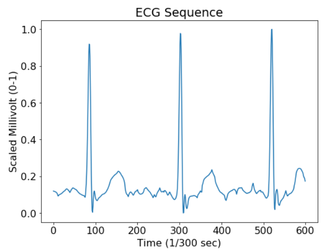

Fig. 4. Example of ECG sequence with length of 600 and normalized between 0 and 1.

Data Augmentation

Data augmentation is a strategy that enables a significant increase in the diversity of data available for training models, without actually collecting new data. Some of the data augmentation techniques used include zero padding, random zero bursts, and random resampling. 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.

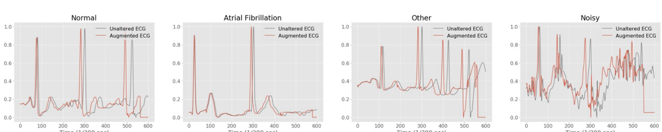

Fig. 5. Example of augmented ECG for each class

Convolutional Neural Network Model

The CNN performs six main function: dimensionalizes inputed ECG data, max pools, convolutes, rectifies, and flattens. The input consists of a 1-dimensional tensor (1x1x600) with a length of 600. The elements are normalized to values between 0 and 1. 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. 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. 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.

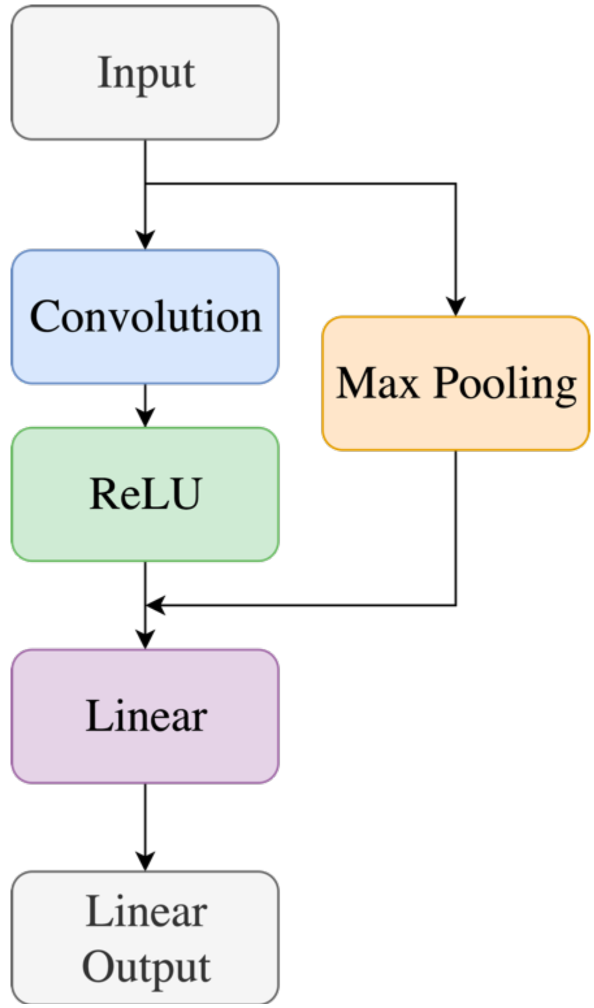

Fig. 6. Flowchart mapping out the Convolutional Neural Network

Evaluation Metrics

Two main evaluation metrics were used in evaluating the CNN: loss, and accuracy. The Cross-Entropy loss function measures how good a prediction from the CNN does in terms of being able to predict the expected outcome (evaluates the outputted values for each class *c*) and aids the CNN in adjusting the weights and bias of a model. Cross-Entropy loss is calculated using the equation:

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

where 4(x, c) is the loss, the parameter x represents the output from the CNN, and c the class number. The total number of classes is N. Furthermore, accuracy measures the correctness of the CNN's prediction with the ground truth. The F1 accuracy is calculated using the equations:

$$\mathrm{A}_n = \frac{2Nn}{(\sum_N + \sum_n)}$$

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

where N is the predicted classification, and n is the reference classification for each class.

Training Set and Validation Set

While training a CNN, it is important to constantly test the model after each epoch to examine if the model is overfitting (memorizing rather than learning) the training data. This is vital to a neural network's success in the real world, as the model will have to identify ECGs that it has never viewed before. To prevent overfitting, techniques like data augmentation can be used at the basic level. However, to be confident the CNN is not overfitting, a validation set can be made to assess the accuracy of the model while training. The validation set was created by partitioning 10% of the preprocessed data.

IV. Training Results

Loss and Accuracy

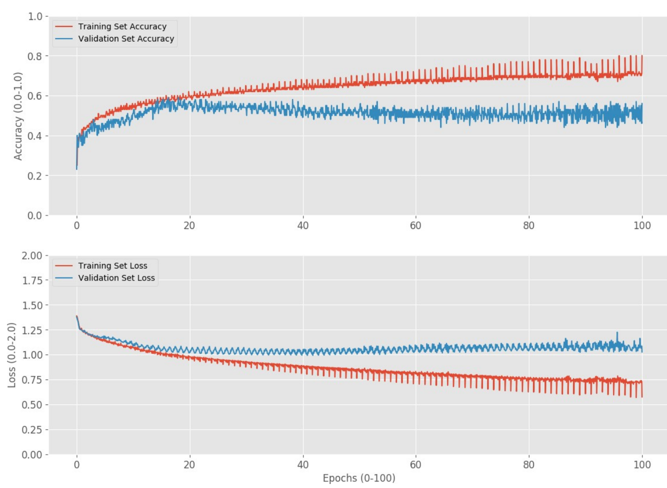

Fig. 7. Graph of the CNN metrics without data augmentation

The graph shows the CNN metrics for a model with 319 hyperparameters, and training without data augmentation. From graph, the CNN's accuracy resembles a logarithmic curve. As the validation set's accuracy approaches a horizontal asymptote at 60% and a point of inflection, the accuracy decreases. Likewise, the loss curve mimics an exponential curve, approaching a minimum loss of 1.0.

This 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.

Confusion Matrix

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

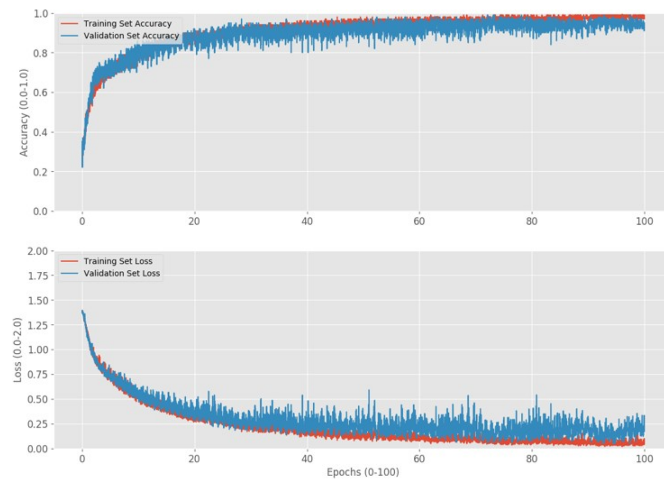

Fig. 8. Graph of the CNN metrics with data augmentation

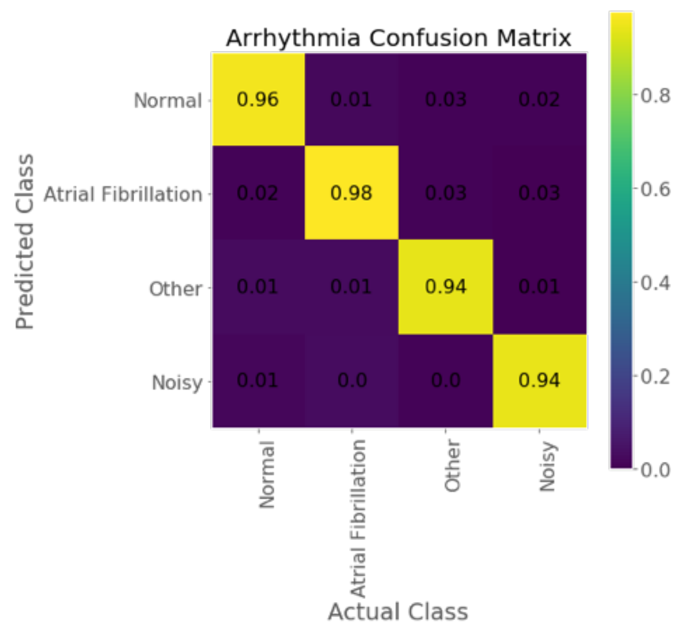

Fig. 9. Confusion matrix on model predictions for each class in the training data

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.

V. 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 anno- tated 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, the model did not learn fast, 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.

VI. Further Exploration and Application

Further explorations include and applications: implementing larger datasets with multiple nodes that record the heart ' s electrical activity simultaneous; such as the Apnea-ECG Database, or the CTU-UHB Intrapartum Cardiotocography Database. Optimizing the time taken to identify an arrhythmia ECG, which allows for faster training and response times. Applying 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. Implementing the CNN model in an ECG reader to autonomously identify arrhythmias in emergencies. Decreaseing the number of misdiagnosis in arrhythmias.

Acknowledgment

I would like to Harrison Kinsley from the Sentdex Youtube channel for creating a Deep Learning series, where I learned most of the basics. I would also like to thank PhyisoNet for graciously proving substantial ECG data publicly, this project would not be possible with their contribution.

REFERENCES

[1] X. Gao, 2019. [Online]. Available: https://www.intechopen.com/online-first/diagnosing-abnormal-electrocardiogram-ecg-via-deep-learning

[2] Awni, 2019. [Online]. Available: https://github.com/awni/ecg

[3] Z. Zhang, F. Duan, J. Sole-Casals, J. Dinares-Ferran, A. Cichocki, Z. Yang, and Z. Sun, "A Novel Deep Learning Approach With Data Augmentation to Classify Motor Imagery Signals," *IEEE Access*, vol. 7, pp. 15 945–15 954, 2019. [Online]. Available: 10.1109/access.2019.2895133;https://dx.doi.org/10.1109/access.2019.2895133

[4] Awerdich, 2020. [Online]. Available: https://github.com/awerdich/physionet

[5] Cvxtz, 2019. [Online]. Available: https://github.com/CVxTz/ECG_Heartbeat_Classification

[6] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional Recurrent Neural Networks for Electrocardiogram Classification," *Cardiology Conference (CinC)*, 2017.

[7] [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[8] B. Pyakillya, N. Kazachenko, and N. Mikhailovsky, "Deep Learning for ECG Classification," *Journal of Physics: Conference Series*, vol. 913, pp. 012 004–012 004, 2017. [Online]. Available: 10.1088/1742-6596/913/1/012004;https://dx.doi.org/10.1088/1742-6596/913/1/012004

[9] M. Alfaras, Soriano, and Silvia, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphy.2019.00103/full

[10] X. R. Zhang, M. Y. Lei, and Y. Li, "An Amplitudes-Perturbation Data Augmentation Method in Convolutional Neural Networks for EEG Decoding," *5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, 2018.

[11] V. Bushaev, 2018. [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[12] N. T. Srinivasan and R. J. Schilling, 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6020177/

[13] P. Cao, X. Li, K. Mao, F. Lu, G. Ning, L. Fang, and Q. Pan, "A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation," *Biomedical Signal Processing and Control*, vol. 56, pp. 101 675–101 675, 2020. [Online]. Available: 10.1016/j.bspc.2019.101675;https://dx.doi.org/10.1016/j.bspc.2019.101675

[14] J. E. Wathen, A. B. Rewers, A. T. Yetman, and M. S. Schaffer, "Accuracy of ECG Interpretation in the Pediatric Emergency Department," *Annals of Emergency Medicine*, vol. 46, no. 6, pp. 507–511, 2005. [Online]. Available: 10.1016/j.annemergmed.2005.03.013;https://dx.doi.org/10.1016/j.annemergmed.2005.03.013

[15] P. Skalski, 2019. [Online]. Available: https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a

[16] J. Brownlee, 2019. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[17] "Heart arrhythmia," *Mayo Clinic*, 2019.

[18] A. Sakai, Y. Minoda, and K. Morikawa, "Data augmentation methods for machine-learning-based classification of bio-signals," *10th Biomedical Engineering International Conference (BMEiCON)*, 2017.

[19] C-Labpl, 2018. [Online]. Available: c-labpl/qrs_detector. Retrievedfromhttps://github.com/c-labpl/qrs_detector

[20] J. Brownlee, 2019. [Online]. Available: https://machinelearningmastery.com/what-is-deep-learning/

[21] X. Gao, 2019.

[22] Krylatov-Pavel, 2019. [Online]. Available: https://github.com/krylatov-pavel/aibolit-ECG

[23] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data Augmentation for Deep Learning-Based Radio Modulation Classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.