Arrhythmia/Heart Monitoring Project Report

Joshua Sweet
Lauren Scott
Sravani Ravula
May 2020

Objective

The objective of this project was to perform an analysis of machine learning and neural network models that perform over heartbeats gathered from Electrocardiographs. This project uses real health informatics data from several parts of the human body recorded through ECG (Electrocardiographs), and mainly targets people with cardiovascular diseases, such as heart attack and congestive heart failure, as they are the leading cause of death in the United States. To do this, an existing database of ECG readings were used along with a few machine learning algorithms to learn how to work with the data and decide what type of algorithm would be suited to develop an accurate prediction.

Tasks

- Analyze the MIT dataset to learn the dataset and the type of annotations it uses along with understanding how to process the data.

- Apply supervised learning, the machine learning task of learning a function that maps an input to an output based on example input-output pairs the data contains.

- Determine the process of feature extraction for the dataset, which is the process of transforming the data values to more meaningful and useful information that can be used in other techniques, such as machine learning. By extracting features about the data the model can predict classifications by relating similar values to each other.

- Apply an SVM based machine learning model to the dataset to get an understanding of how prediction and classification can be applied to the set.

- Implement more advanced neural networking tests to determine which model returns the most accurate result. The neural network is optimal because It runs multiple trials across many epochs to evaluate a final result.

- Future plans for this project also included using Django and developing a front end, so even the non-CS users can use it and get the predicted diseases.

Data Set

The MIT-BIH arrhythmia database is an open-source dataset that provides standard investigation material for the detection of heart arrhythmia. The datasets are stored with .atr, .dat, .hea, and .xws extensions with about 73.5 MB file size per dataset. Each recording was digitized at 360 samples per second for each channel with an 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database. Data is stored in two variations, analog and digital. Analog is stored in floating-point values in the +/- 5 mV range. Digital is a conversion from analog and is stored in an 11-bit value.

Cite: https://www.physionet.org/content/mitdb/1.0.0/

Software & Libraries

- Python 3.7.
- Anaconda.
- Jupyter Notebook – execution environment to visualize data.
- Scipy (matplotlib, NumPy, pandas) - Python libraries to analyze & graph data.
- Wfdb – python library used to process waveform database signals
- Tensorflow and SVM – Machine Learning applications used to predict classifications (neural network).

WFDB Library

The wfdb library converts from analog to digital with simple algebra, and vice versa. Two values are required, the gain and baseline and most of the files used digital singles using pickle. (Pickling is a way to convert a python object into a character stream).

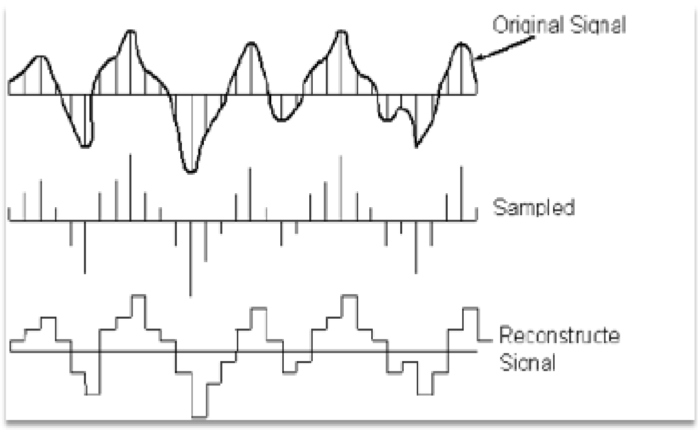

Figure 1: processing signal (raw data) to digital

Calculation of Analog to Digital: (ANALOG * GAIN) + BASELINE

Reverse : (VALUE - BASELINE) / GAIN

Due to a large number of files in the raw data set (about 500 MB in total), a compressed version of the dataset was used for the neural network models, which reduced the file size from 500 MB to 44 MB.

Data Extraction

Feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved.

- It is the act of processing the retrieved data from the data set for analysis since data may contain formatting or just be raw digit values..

- Classification models require you to label your training set before you train it as data that is extracted can be labeled or unlabeled.

- Some of the lead types are MIL (right arm) and V5, V4, V2, and, V1 which is the left side of the body.

- Used Sklearn and Matpolit to visualize each type over the period of time to better understand the behavior of the signal for the future process of feature selection.

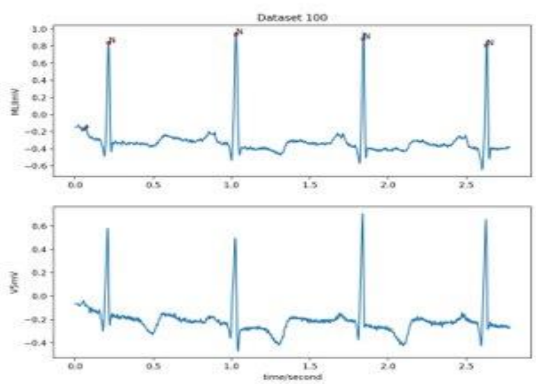

Figure 1: MLV and V5 Signal over time

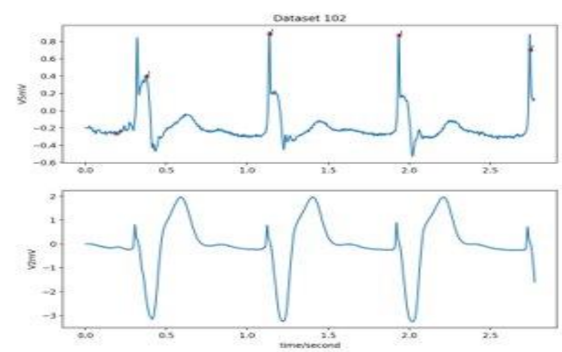

Figure 2: V2 and V5 Signal over time

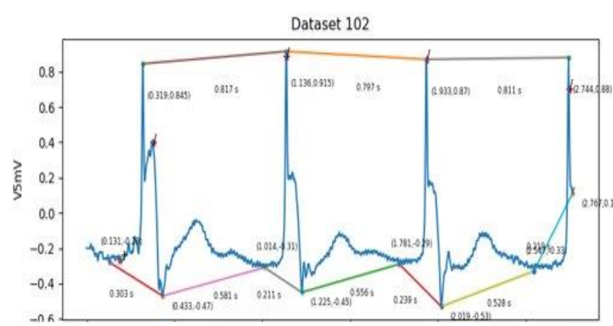

Figure 3: Peaks over time

The raw data contains timestamps of noteworthy rhythms. Most are located at the peak of a beat. This can be taken advantage of by parsing the data into 1-second chunks, with the peak of the noteworthy beat at the center. With this, approximately 220000 1-second samples can be obtained, each labeled with the respective rhythm type.

Models

When performing learning on the set, one machine learning model and three neural network models were used. A distinction should be made between the two such that:

1. Machine Learning
   Machine learning is the process of comparing data towards mathematical functions to construct another function that is used to predict the data by either classifying it or predicting new values for it. This is done by figuring out the set of rules that apply to the dataset, in which the rules are constructed from the data and their respective answers. The end goal is that given a predicted set of rules, data can be passed to it and it can produce an answer that represents what the actual value might be.

2. Neural Networking
   A neural network is a set of algorithms which are modeled loosely off of the human brain that are designed to recognize patterns to make predictions of inputs with similar patterns. Neural networks do this by performing iterations over data and running each iteration through a set of mathematical algorithms known as layers, in which each layer performs a mathematical operation over the set of data, which can range from actual prediction or just simply normalizing the data.

The main overall difference is that a machine learning model predicts by using a mathematical function generated from the data such that given an input it will produce an output, while a neural network predicts by comparing iterations on the set of data and making a prediction based on the similarity between each set tested.

1. Support Vector Machine (SVM)

The objective of a Support Vector Machine model is to generate a hyperplane in N-dimensional space where N is the number of classifiers for the model. When training is performed on the model, it will find the optimal hyperplane and then fit the testing values to that plane, in which predictions not resulting in the hyperplane are marked as incorrect.

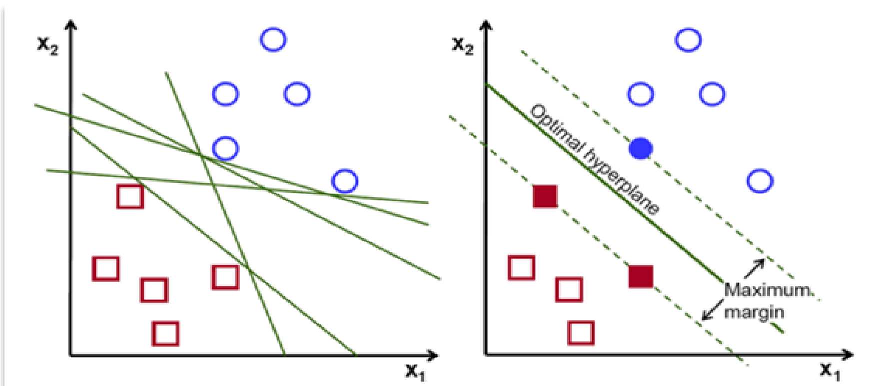

Figure 4. Support Vector Machine representation.[1]

2. Convolutional Neural Network (CNN)

As the first neural network tested, a Convolutional Neural Network works by receiving an input and running the input through linearly connected layers until it reaches the end. By acting in a linear manner, it means the prediction algorithm is not susceptible to large changes based on the data it receives.

3. Combined Densely Connected Neural Network (cDNN)

cDNN functions by creating a second neural network model for each input it receives, which is built using mini-networks, and then maps the input to that model created. After each input has been put through this pattern, it then combines all the results by mapping them to a final layer. This is done to allow the prediction algorithm to change based on the behavior of the input it receives.

4. Recurrent Neural Network (RNN)

The function of RNN is to establish a directed graph between each layer in the network, such that it takes an input and has it transition between layers until a decision is made to bring it to the final state. This is done to allow for dynamic processing of recognizing patterns.

5. Discrete Wavelet Transform

Discrete Wavelet Transform is a sub-form of cDNN that splits the input into multiple sub-inputs before performing on it. When each of these inputs goes through the cDNN network, each network works independently of each other and as a result performs predictions on just that set of input received. The results afterwards are combined and sent through a final dense layer to form a final prediction using the results from each performance of cDNN. This works to assure detailed learning on the input data being given to the model.

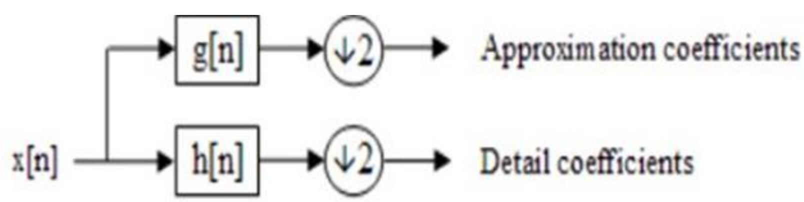

Figure 5. Discrete Wavelet Transform with a 1 level filter. [2]

Here, x[n] is the original input while g[n] and h[n] are the sub-inputs, where g[n] represents approximation coefficient (inputs) and h[n] are the more detailed coefficients. The ↓2 means that the data is half of the original input. The values of h[n] for each level is where the learning will be applied.

The term 'level' refers to how many times discrete wavelet will apply itself on g[n] to locate as many detailed coefficients as it can. For each additional level, the input will be g[n] and will split once again into half based on the same values. Level 1 means the process is applied only to the initial input.

Results

1. ## SVM Model

   It was found that the SVM model returned 89% correct classification results. Below are two figures: a screenshot of the output after running the SVM model, and a confusion matrix of the results. The confusion matrix shows that most of the results reside on the diagonal line, indicating correctness. The brighter the color of the square, the more results it represents.

	precision	recall	f1-score	support
0	0.80	0.88	0.84	997
2	0.96	0.97	0.96	2046
3	0.97	0.98	0.97	1794
4	0.60	0.10	0.18	29
5	0.79	0.92	0.85	1755
6	0.91	0.74	0.82	195
7	0.64	0.32	0.42	22
8	0.90	0.78	0.84	641
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	30
11	0.73	0.62	0.67	65
12	0.99	0.99	0.99	1716
13	0.33	0.10	0.15	10
14	0.41	0.11	0.18	179
16	0.40	0.07	0.12	29
22	0.94	0.76	0.84	105
28	0.53	0.64	0.58	326
31	0.90	0.35	0.50	107
32	0.00	0.00	0.00	1
33	0.00	0.00	0.00	1
34	0.00	0.00	0.00	5
37	0.93	0.66	0.77	41
38	0.89	0.74	0.81	253
accuracy			0.89	10348
macro avg	0.59	0.47	0.50	10348
weighted avg	0.88	0.89	0.88	10348

Figure 6. A printout showing the results after running the SVM model.

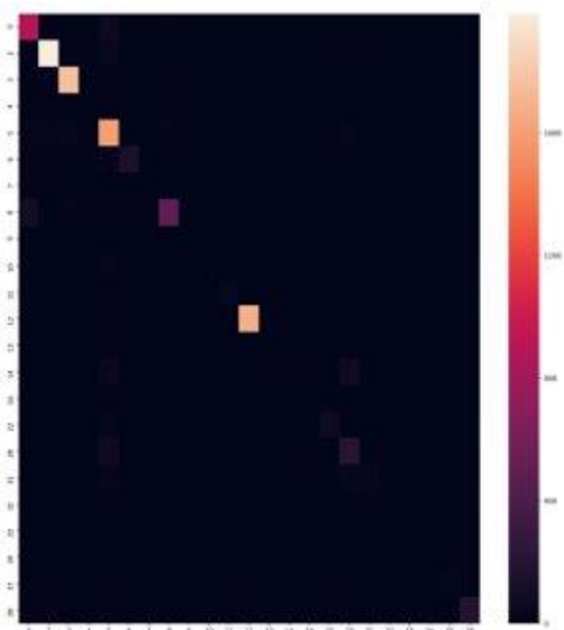

Figure 7. A confusion matrix

To understand what the classifier numbers represent in terms of heart annotations see the annotation table below. (Fig 6 & 7).

Label Store	Symbol	Description
0	0	Not an actual annotation
1	N	Normal beat
2	L	Left bundle branch block beat
3	R	Right bundle branch block beat
4	A	Aberrated atrial premature beat
5	V	Premature ventricular contraction
6	F	Fusion of ventricular and normal beat
7	J	Nodal (junctional) premature beat
8	A	Atrial premature contraction
9	S	Premature or ectopic supraventricular beat
10	E	Ventricular escape beat
11	J	Nodal (junctional) escape beat
12	/	Paced beat
13	Q	Unclassifiable beat
14	~	Signal quality change
16	|	Isolated QRS-like artifact
18	S	ST change
19	T	T-wave change
20	*	Systole
21	D	Diastole
22	"	Comment annotation
23	=	Measurement annotation
24	P	P-wave peak
25	B	Left or right bundle branch block
26	^	Non-conducted pacer spike
27	T	T-wave peak
28	+	Rhythm change
29	U	U-wave peak
30	?	Learning
31	!	Ventricular flutter wave
32	[	Start of ventricular flutter/fibrillation
33	]	End of ventricular flutter/fibrillation
34	e	Atrial escape beat
35	n	Supraventricular escape beat
36	@	Link to external data (aux_note contains URL)
37	x	Non-conducted P-wave (blocked APB)
38	f	Fusion of paced and normal beat
39	(	Waveform onset

This table applies to both the SVM model and the neural networks. The classification given to the waveform based on the annotations means that annotation specifically is a common reoccurring annotation and therefore an overall issue. (Unless a normal classification was predicted) Possible change could be to remove general comment annotations from potential classification

2. Neural Network

The neural networks overall had better performance than the SVM model, with greater accuracy on average. However, some models resulted in a final accuracy lower than the SVM model. For example, CNN using 10 K-folds (trials) and 20 epochs resulted in a final accuracy of approximately 78 percent, a full 11 percentage points lower than the SVM model.

Out of the various neural network trials performed, the cDNN model with Discrete Wavelet Transform was found to be the most accurate model on average. When running with 100 epochs, it produced accuracy values greater than 90%.

3. Discrete Wavelet Transform

The cDNN model with Discrete Wavelet Transform resulted in accuracy levels of 90% or greater for three of the four test cases performed. When it was run with less than 100 epochs, accuracy values were lower, but this is to be expected as less iterations were performed this way which means that the classification was less sensitive.

A table summarizing the results from running the cDNN model at 100 epochs is shown below.

	adam 2	adam 39	adadelta 2	adadelta 39
Trial 1	Loss: 0.411 Acc: 0.947	Loss: 0.920 Acc: 0.922	Loss: 0.252 Acc: 0.909	Loss: 0.622 Acc: 0.842
Trial 2	Loss: 0.356 Acc: 0.950	Loss: 1.143 Acc: 0.924	Loss: 0.261 Acc: 0.906	Loss: 0.590 Acc: 0.855
Trial 3	Loss: 0.447 Acc: 0.952	Loss: 1.114 Acc: 0.926	Loss: 0.241 Acc: 0.911	Loss: 0.593 Acc: 0.856
Trail 4	Loss: 0.319 Acc: 0.958	Loss: 1.124 Acc: 0.920	Loss: 0.272 Acc: 0.900	Loss: 0.655 Acc: 0.835
Trail 5	Loss: 0.370 Acc: 0.959	Loss: 0.977 Acc: 0.925	Loss: 0.261 Acc: 0.901	Loss: 0.615 Acc: 0.842
Avg.	Loss: 0.38 Acc 95.30	Loss: 1.06 Acc 92.34	Loss: 0.26 Acc 90.54	Loss: 0.62 Acc 84.64

Figure 8. A table which shows the results from running the cDNN model with Discrete Wavelet Transform.

Conclusion

After testing different models and based on the research, of the five models tested on the MIT dataset, cDNN with Discrete Wavelet Transform was able to produce the most accurate consistent final results. As this is an ongoing project there can be more research done for cleaning the noise produced by the ECG signals and work on the possible future classification using the different model in machine learning to predict potential with the actual data when produced by the sponsors.

Works Cited

Ippolito, Pier Paolo. “SVM: Feature Selection and Kernels.” *Medium*, Towards Data Science, 12 Sept. 2019, towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c.

“Discrete Wavelet Transform.” *Wikipedia*, Wikimedia Foundation, 22 Apr. 2020, en.wikipedia.org/wiki/Discrete_wavelet_transform.

Alal, Ethar. “What component in a computer plays the role of an analog to digital converter for electricity that is coming out of the wall socket?” *Quora.* 29 Dec, 2016, https://www.quora.com/What-component-in-a-computer-plays-the-role-of-an-analog-to-digital-converter-for-electricity-that-is-coming-out-of-the-wall-socket