ECG Arrhythmia Detection with MIT-BIH Data set

By Alton Lavin D’souza
Varun Dave
Malikarjun Modi

Aim of the project

- Classify the following 5 classes
  - Normal (N) - 73439
  - Paced Beat (/) - 8068
  - Right Bundle Branch Block Beat (R) - 7255
  - Left Bundle Branch Block (L) - 6793
  - Premature Ventricular Beat (V) - 3619
- For ML:
  - N=1
  - v=2
  - /=3
  - L=4
  - R=5
- For DL
  - /=0
  - L=1
  - N=2
  - R=3
  - V=4

Methods

- Data Cleaning and Preprocessing
  - Fourier Transform
  - Low Pass filter
  - High Pass filter
  - R peak detectors
- Deep Learning Models
  - ANN - 99% accuracy
  - CNN - 99% accuracy
  - LSTM - 99% accuracy
- Machine Learning Models
  - Logistic Regression - 84% accuracy
  - KNN - 99% accuracy
  - Decision Tree - 98% accuracy
  - Random Forest - 99% accuracy
  - SVM -98% accuracy
  - Gaussian Naive Bayes - 70% accuracy

- Computing Resources: CPU

Method - Logistic Regression Results

- Accuracy: 84%
- Confusion Matrix:
- Hyperparameter: multinomial

Class	N	V	/	L	R
N	14562	27	1	66	32
V	647	476	22	112	101
/	6	3	703	12	0
L	1051	26	1	534	2
R	981	14	0	23	433

Method - Logistic Regression Results

- Identifying L, logistic regression did the worst
- Adding more data might improve its classification
- The same can be said for R
- The result for / with a small amount of data is astonishing we need more testing data

Class/Metric	Precision	recall	f1-score	support
N	0.84	0.99	0.91	14688
V	0.87	0.35	0.50	1358
/	0.97	0.97	0.97	724
L	0.71	0.33	0.45	1614
R	0.76	0.30	0.43	1451
accuracy			0.84	19835
Macro avg	0.83	0.59	0.65	19835
Weighted avg	0.83	0.84	0.81	19835

Method - KNN Results

- Accuracy: 99%
- Hyperparameter: k=5

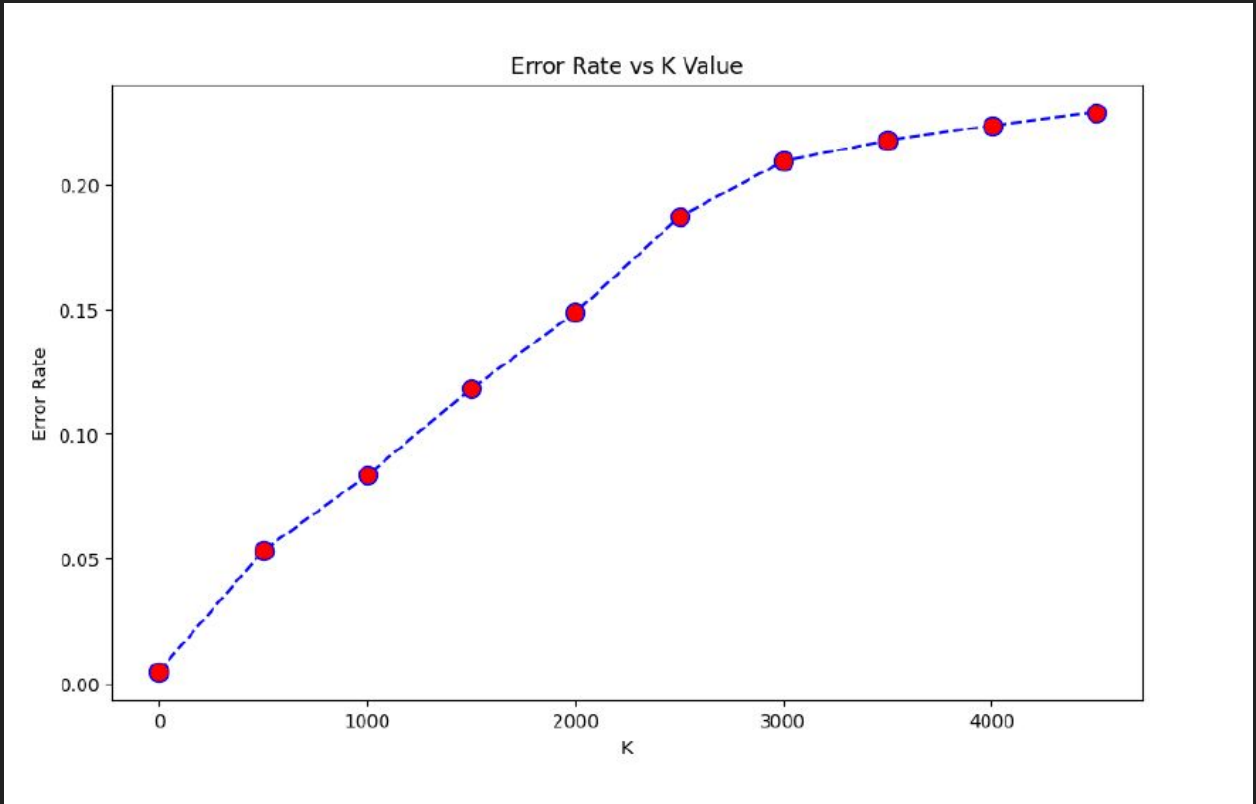
Class	N	V	/	L	R
N	14659	20	2	3	4
V	47	1288	1	18	4
/	0	0	723	1	0
L	7	4	0	1603	0
R	6	0	0	1	1444

Method-KNN Results

- Overall the model seems to be performing well
- The K value was chosen using the elbow method
- Classifying V is still a problem

Class/Metric	Precision	recall	f1-score	support
N	1.00	1.00	1.00	14688
V	0.98	0.95	0.96	1358
/	1.00	1.00	1.00	724
L	0.99	0.99	0.99	1614
R	0.99	1.00	0.99	1451
accuracy			0.99	19835
Macro avg	0.99	0.99	0.99	19835
Weighted avg	0.99	0.99	0.99	19835

Method - KNN Results



Method - Decision Tree Results

- Accuracy: 98%
- Hyperparameter: criterion=entropy and splitter=random

Class	N	V	/	L	R
N	14585	60	2	21	20
V	66	1255	5	16	16
/	4	4	715	1	0
L	22	17	0	1570	5
R	27	3	0	2	1419

Method - Decision Tree Results

- Classifying V is still a problem
- Overall the model performs well

Class/Metric	Precision	recall	f1-score	support
N	0.99	0.99	0.99	14688
V	0.94	0.92	0.93	1358
/	0.99	0.99	0.99	724
L	0.98	0.97	0.97	1614
R	0.97	0.98	0.97	1451
accuracy			0.99	19835
Macro avg	0.97	0.97	0.97	19835
Weighted avg	0.99	0.99	0.99	19835

Method - Random Forest

- Accuracy: 0.99
- Hyperparameter: criterion=gini

Class	N	V	/	L	R
N	14667	15	1	3	2
V	59	1289	2	5	3
/	2	1	721	0	0
L	19	14	0	1581	0
R	16	1	0	2	1432

Method - Random Forest Results

- Classifying V is still a problem
- Overall the model is still a problem

Class/Metric	Precision	recall	f1-score	support
N	0.99	1.00	1.00	14688
V	0.98	0.95	0.96	1358
/	1.00	1.00	1.00	724
L	0.99	0.98	0.99	1614
R	1.00	0.99	0.99	1451
accuracy			0.99	19835
Macro avg	0.99	0.98	0.99	19835
Weighted avg	0.99	0.99	0.99	19835

Method SVM Results

- Accuracy: 98
- Hyperparameter: C=0.1 and gamma=1

Class	N	V	/	L	R
N	14608	39	0	32	9
V	102	1228	2	14	12
/	1	1	722	0	0
L	95	17	0	1502	0
R	118	2	0	1	1330

Method SVM Result

- Classifying V is still an issue but this time we see some issues with L
- Overall the model performs well

Class/Metric	Precision	recall	f1-score	support
N	0.98	0.99	0.99	14688
V	0.95	0.90	0.93	1358
/	1.00	1.00	1.00	724
L	0.97	0.93	0.95	1614
R	0.98	0.92	0.95	1451
accuracy			0.98	19835
Macro avg	0.98	0.95	0.96	19835
Weighted avg	0.98	0.98	0.98	19835

Method - Gaussian Naive Bayes

- Accuracy: 0.71%

Class	N	V	/	L	R
N	10325	807	149	2660	747
V	38	748	174	324	74
/	1	44	678	1	0
L	217	91	6	1291	9
R	341	13	0	53	1044

Method - Gaussian Naive Bayes

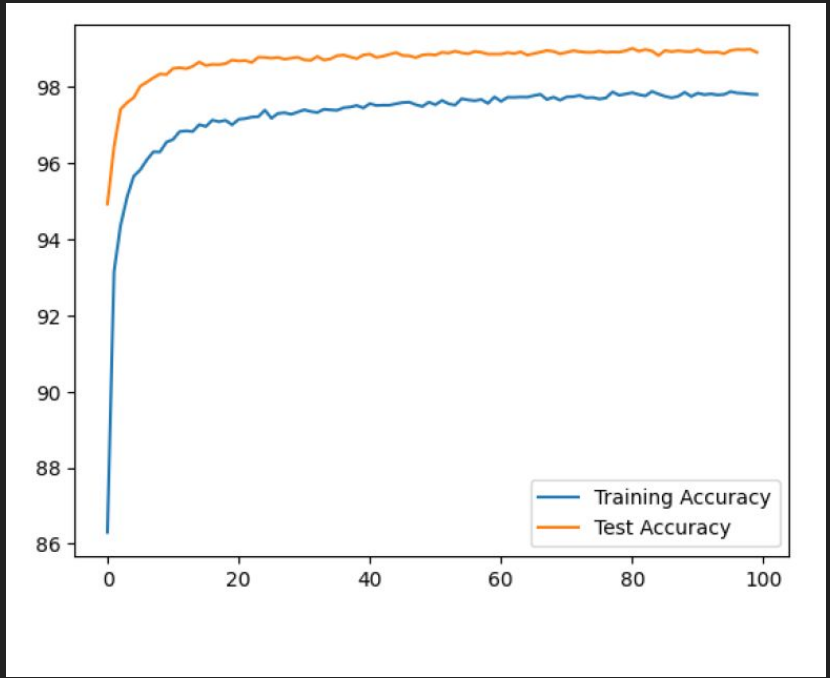
- Doesn’t perform well
- Needs more data to work looks like in each class

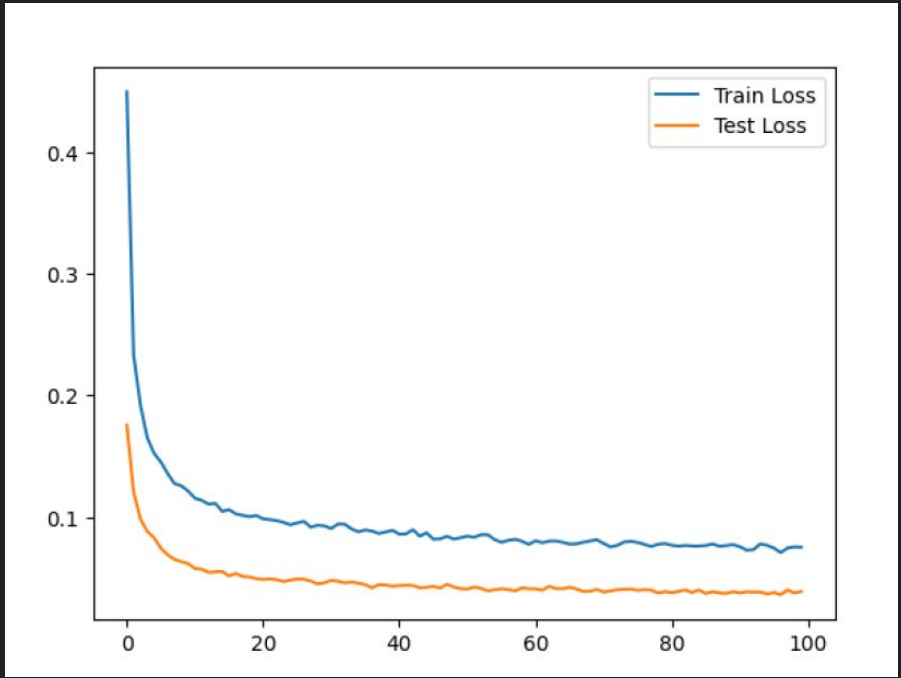
Class/Metric	Precision	recall	f1-score	support
N	0.95	0.70	0.81	14688
V	0.44	0.55	0.49	1358
/	0.67	0.94	0.78	724
L	0.30	0.80	0.43	1614
R	0.56	0.72	0.63	1451
accuracy			0.71	19835
Macro avg	0.58	0.74	0.63	19835
Weighted avg	0.82	0.71	0.74	19835

Method - ANN Results

- Architecture consists of the linear layers with an input size of 200,100 ,50 and 5
- Each linear layer has Relu activation function except for the last layer
- Each layer is followed by a Drop Out layer to help with Regularization and prevent overfitting
- Accuracy: 99 %
- For one 200 samples of ECG reading the Computation Complexity is 25.55KMac and 51.1 KFlops
- Parameters: 25.41K
- Number of Epochs:100
- Training Duration: 761 seconds
- Test Duration: 1 second

Method - ANN Results





Method - ANN Results

Class	/	L	N	R	V
/	723	0	0	0	1
L	0	1575	12	0	7
N	0	23	14637	14	60
R	0	0	5	1436	1
V	1	16	34	1	1289

Method ANN Model

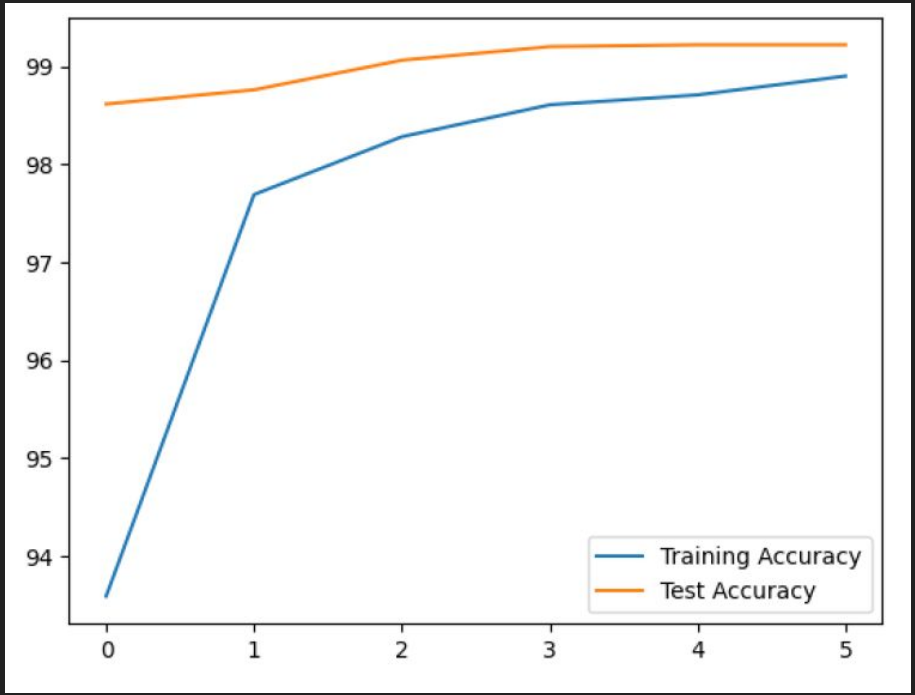
- L and V need more data V test data is not enough
- Overall the model does well

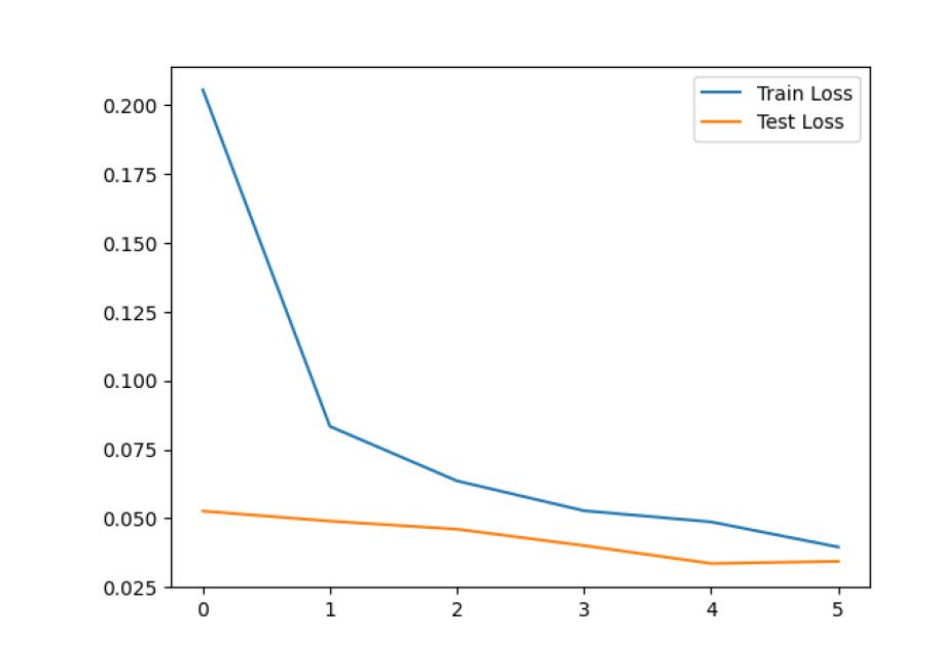
Class/ Metric	Precision	recall	f1-score	support
/	1.00	1.00	1.00	724
L	0.98	0.99	0.98	1594
N	1.00	0.99	0.99	14734
R	0.99	1.00	0.99	1442
V	0.95	0.96	0.96	1341
accuracy			0.99	19835
Macro avg	0.98	0.99	0.98	19835
Weighted avg	0.99	0.99	0.99	19835

Method CNN Model

- One Convolution layer has 2 conv1d layers with Relu activation function followed by a maxpool layer and batchnorm layer
- This pattern is repeated 2 times and is then followed by a flatten layer, followed by 3 linear layers with each later having a dropout layer and relu activation function . The last liner doesn’t have a dropout or activation function and is used to give us the predictions
- Accuracy:99%
- Number of parameters: 2.47 M
- Computation Complexity for 1 reading of size 200 samples: 8.81 MMac and 17.62 MFlops
- Epochs:6
- Test Duration:12 seconds
- Training Duration:840 Seconds

Method-CNN Model





Method - CNN

Class	/	L	N	R	V
/	723	0	2	0	0
L	0	1594	5	0	17
N	0	10	14659	7	70
R	0	0	5	1444	1
V	1	10	17	0	1270

Method CNN Model

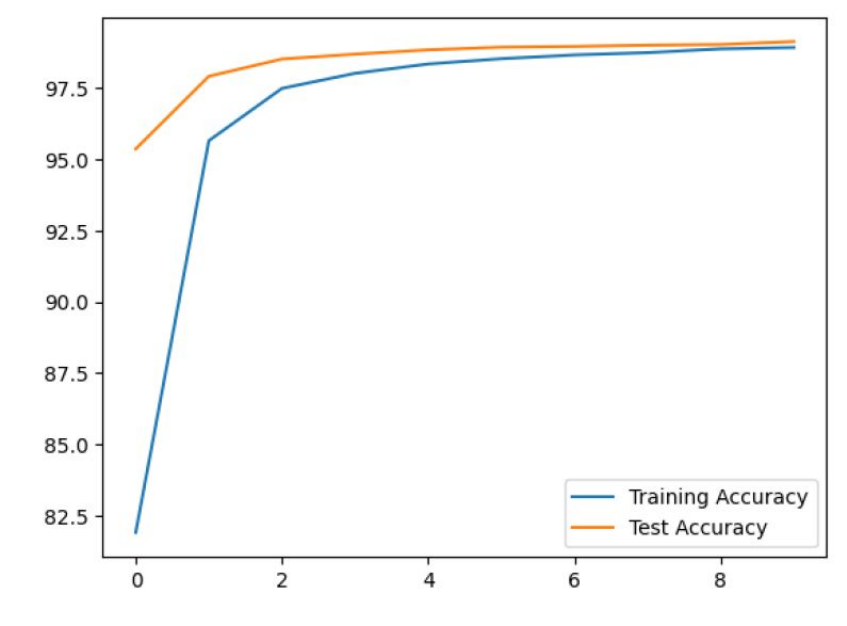
- Some Problem classifying V
- Model take some time to train hence small epoch used
- Time to Train and test can be attributed to the use of convolution layers

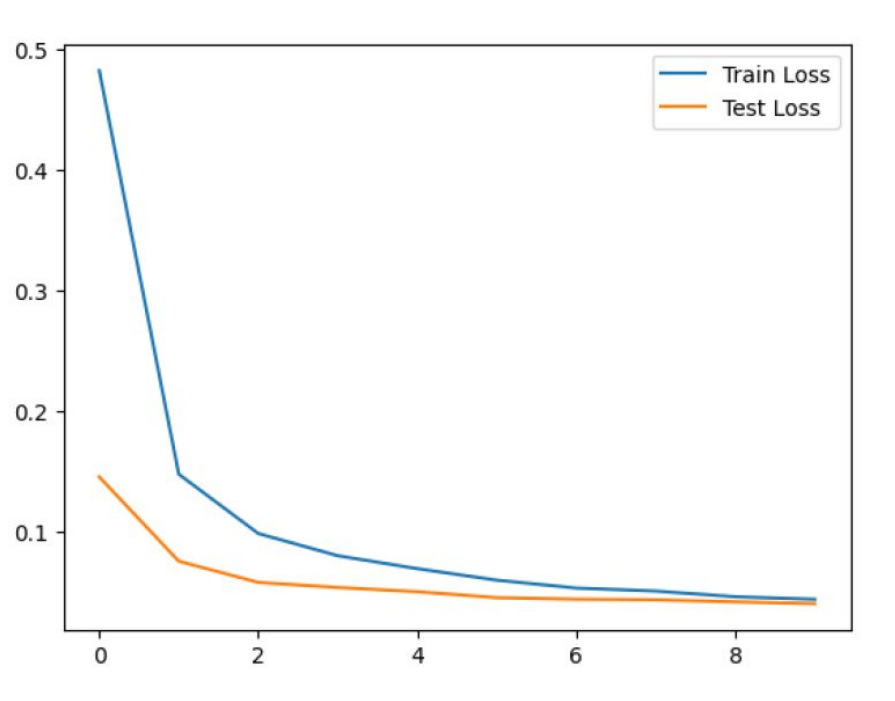
Class/ Metric	Precision	recallc	f1-score	support
/	1.00	1.00	1.00	725
L	0.99	0.99	0.99	1616
N	1.00	0.99	1.00	14746
R	1.00	1.00	1.00	1450
V	0.94	0.98	0.96	1298
accuracy			0.99	19835
Macro avg	0.98	0.99	0.99	19835
Weighted avg	0.99	0.99	0.99	19835

Method LSTM Model

- One LSTM layer with input size of 200 and hidden size of 50
- 2 linear layers one is used with relu and dropout while the other is the final layer to produce outputs
- Accuracy: 99%
- Epochs:10
- Training time: 110
- Testing Time: 1 second
- For a single 200 sample reading the computational complexity is 73.23KMac and 146.46 Flops
- Parameters: 72.2K

Method LSTM Model





Method - LSTM Model

Class	/	L	N	R	V
/	723	0	1	0	0
L	0	1587	19	0	21
N	0	12	14637	9	56
R	0	0	13	1438	3
V	1	15	18	4	1278

Model - LSTM Model

- Uses less parameters for learning
- Classifying V is still an issue
- Overall the model does well

Class/ Metric	Precision	recallc	f1-score	support
/	1.00	1.00	1.00	724
L	0.98	0.98	0.98	1627
N	1.00	0.99	1.00	14714
R	0.99	0.99	0.99	1454
V	0.94	0.97	0.96	1316
accuracy			0.99	19835
Macro avg	0.98	0.99	0.98	19835
Weighted avg	0.99	0.99	0.99	19835

Inferencess

- Except for Logistic Regression and Gaussian Naive Bayes all the models are performing well
- However, since a shallow model like KNN has an accuracy of 99%, it might be better to lean to this model.
- One con is evaluation might take sometime but since we are using 5 neighbours it should be fine
- Most Testing time with 16000+ data takes about 1 sec
- One other preprocessing that was performed was to leverage Oversampling and Under Sampling techniques, but this seemed to lower the accuracy in general.