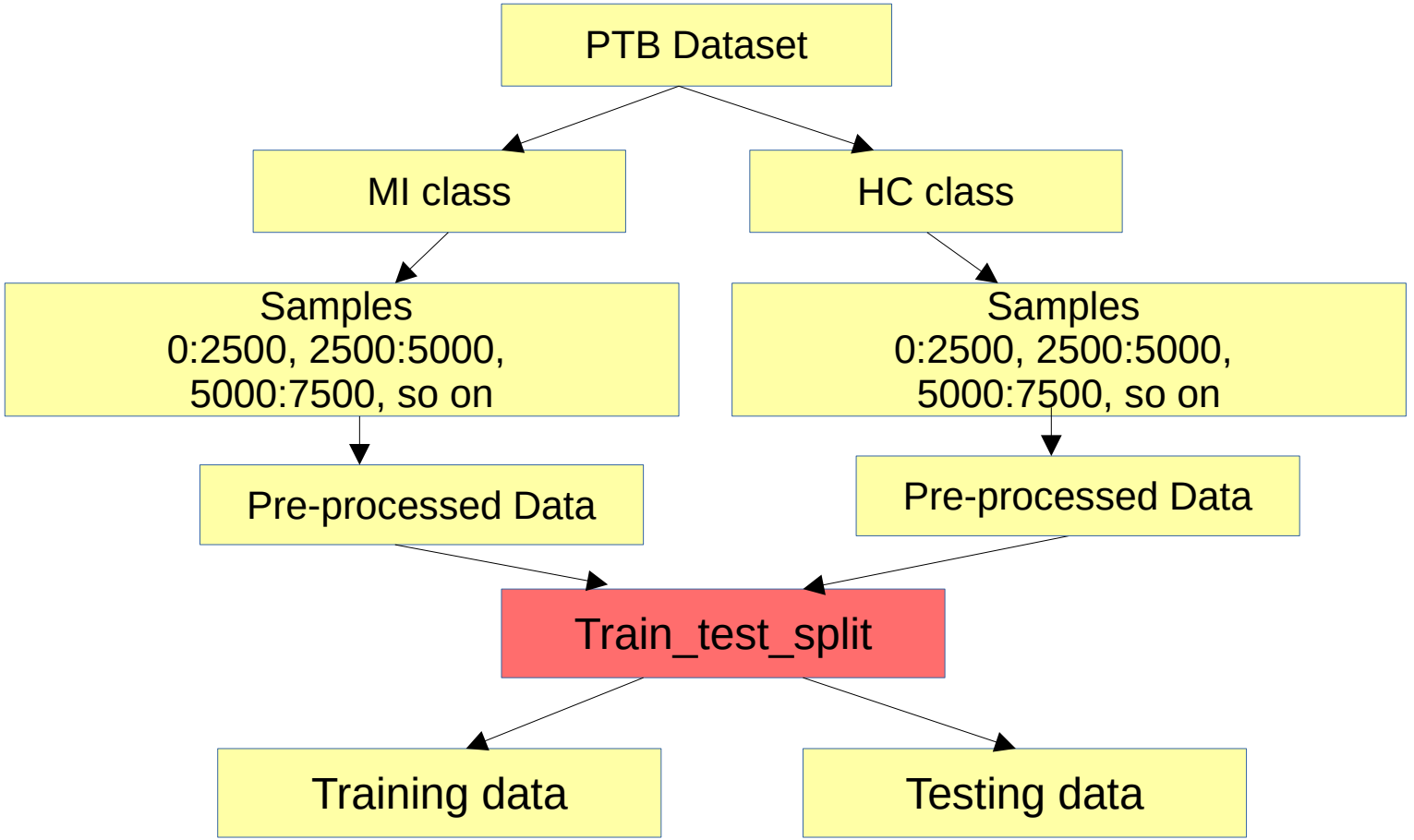
Problem: Very high accuracy using DNN

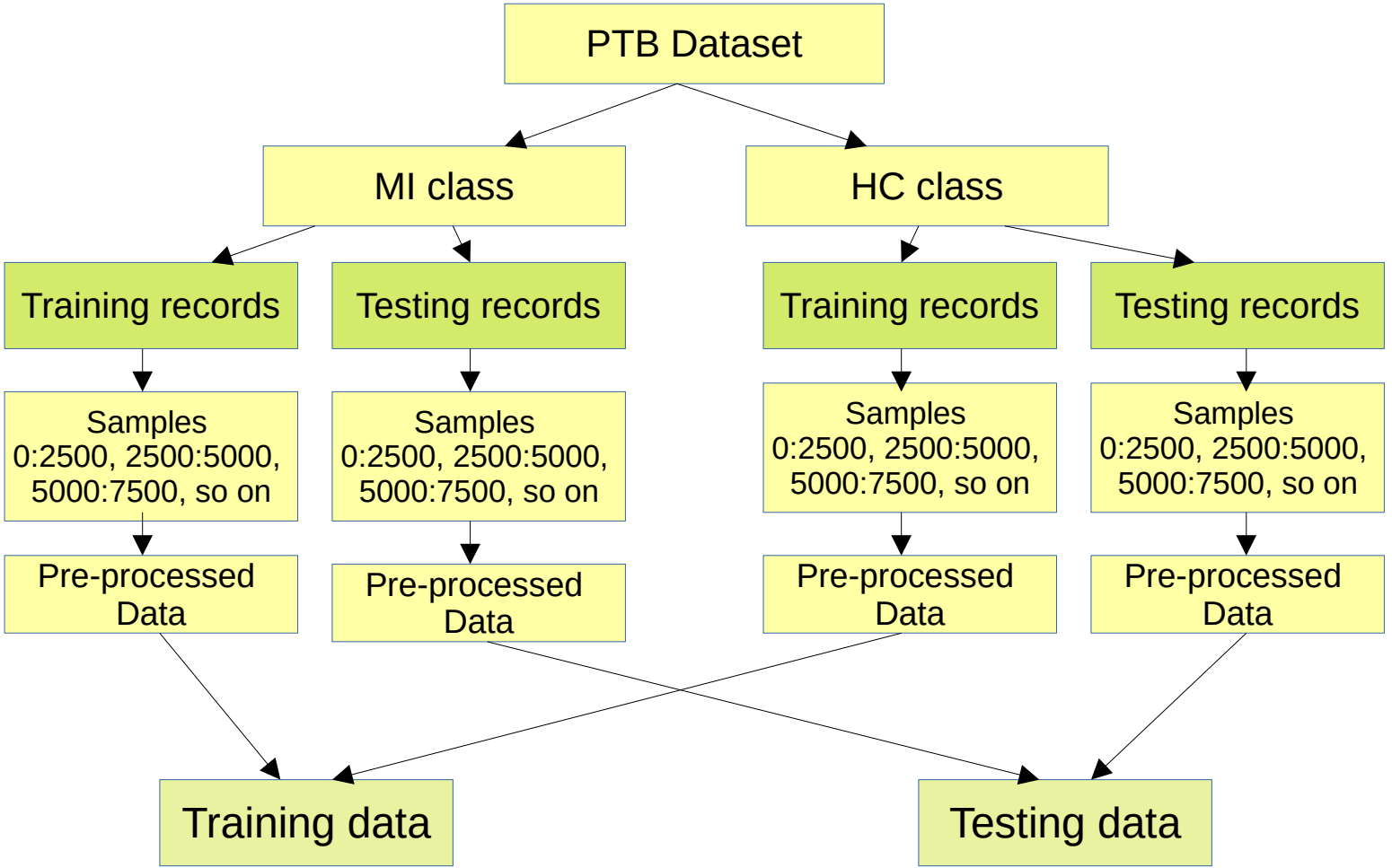
Reason:-

The approach of formation of dataset was not correct. This is causing over-fitting of model during training.

Previous Approach



New Approach



Dataset in table format

	Actual Records	Train Records	Test Records	Train Dataset	Test Dataset
MI	148	118	30	5318	1372
HC	54	42	12	1946	555
Total	202	160	42	7264	1927

Sampling Frequency of Dataset:- 1000 Hz
Downsampled to:- 250 Hz

So, each record has dimension (625, 12)
[2.5 sec* 250Hz; 12 leads]

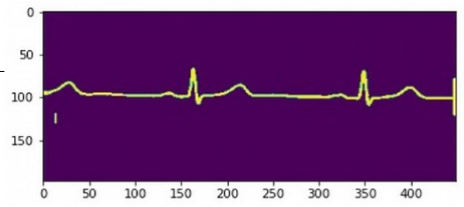
```
(7264, 625, 12)
(7264,)
(1927, 625, 12)
(1927,)
Total number of unique combinations of diagnosis(TRAINING): 7264
MI Records: 5318
HC Records: 1946
Total number of unique combinations of diagnosis(TESTING): 1927
MI Records: 1372
HC Records: 555
```

Create Images and then read them to retrieve the data

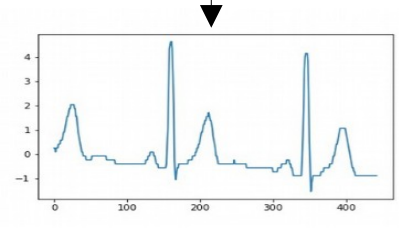
Testing Data/ Training Data

Create .jpg images

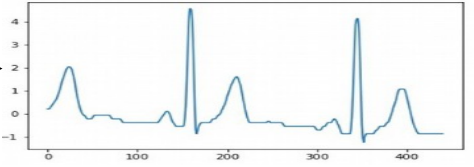
Image Processing



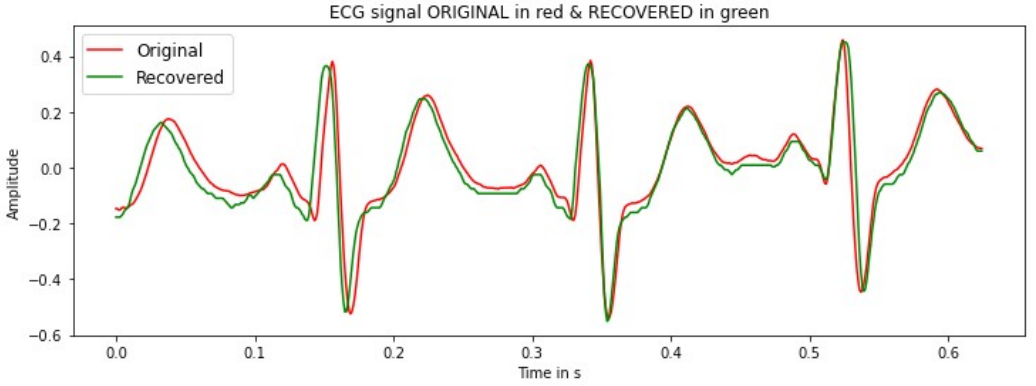
Binary image to plot

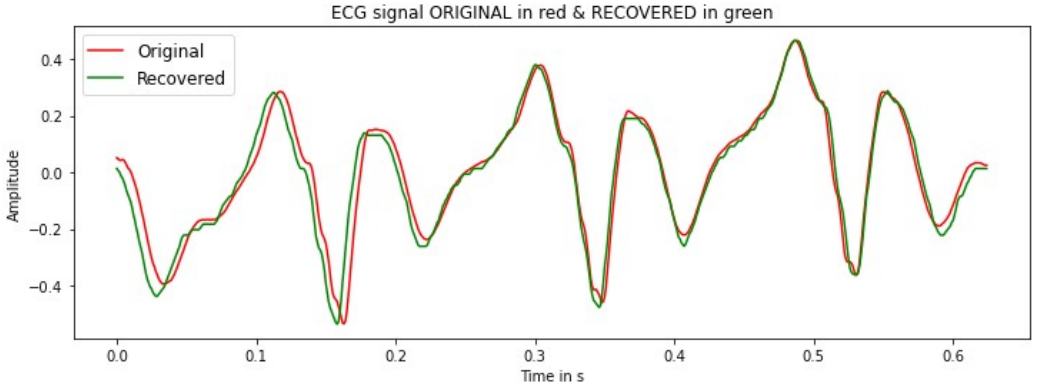


Smooth/Filter out any defects



Objective 2: Visual Comparison





Objective 3:Quantitative Comparison

- For I = 0^{th} record,
  - Mean Square Error:- 0.003045019
  - Mean Absolute Error:- 0.045140551
  - WPRD:-

WPRD: (6+1 levels along rows as cA6 cD6 cD5 cD4 cD3 cD2 cD1)*(12 leads along columns)												
cA6	0.431,	0.216,	0.392,	0.161,	0.423,	0.279,	0.270,	0.376,	0.400,	0.453,	0.204,	0.308,
cD6	0.235,	0.304,	0.337,	0.306,	0.241,	0.272,	0.371,	0.292,	0.364,	0.488,	0.433,	0.451,
cD5	0.625,	0.396,	0.588,	0.499,	0.561,	0.435,	0.627,	0.386,	0.567,	0.725,	0.521,	0.513,
cD4	0.798,	0.605,	0.864,	0.572,	0.786,	0.793,	0.805,	0.593,	0.708,	0.903,	0.657,	0.659,
cD3	1.170,	1.080,	0.835,	0.793,	0.924,	1.024,	1.206,	0.733,	1.101,	1.257,	0.884,	0.966,
cD2	1.040,	1.615,	1.269,	1.265,	1.151,	1.319,	1.158,	0.944,	1.353,	1.333,	0.997,	1.115,
cD1	0.657,	1.428,	0.707,	0.860,	0.580,	0.718,	0.689,	0.508,	0.783,	2.714,	0.910,	1.046,

Conclusion from WPRD calculation

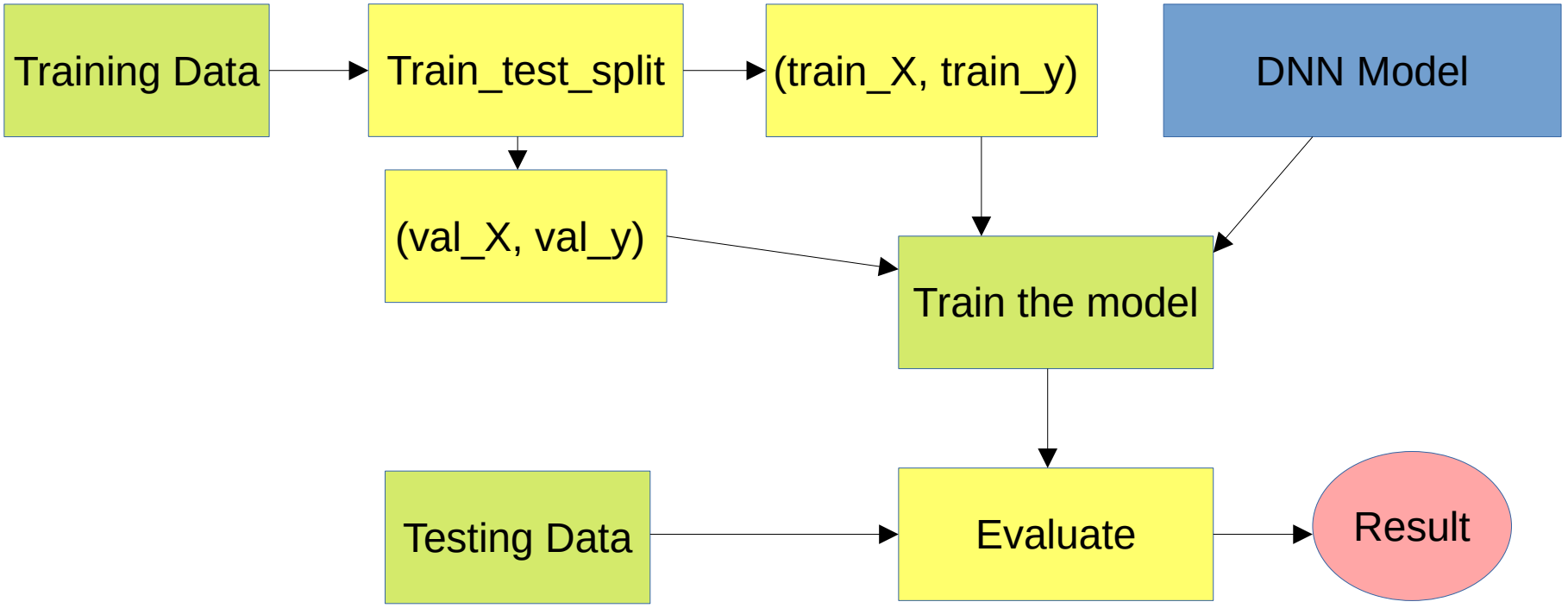
WPRD: (6+1 levels along rows as cA6 cD6 cD5 cD4 cD3 cD2 cD1)*(12 leads along columns)

cA6	0.431,	0.216,	0.392,	0.161,	0.423,	0.279,	0.270,	0.376,	0.400,	0.453,	0.204,	0.308,
cD6	0.235,	0.304,	0.337,	0.306,	0.241,	0.272,	0.371,	0.292,	0.364,	0.488,	0.433,	0.451,
cD5	0.625,	0.396,	0.588,	0.499,	0.561,	0.435,	0.627,	0.386,	0.567,	0.725,	0.521,	0.513,
cD4	0.798,	0.605,	0.864,	0.572,	0.786,	0.793,	0.805,	0.593,	0.708,	0.903,	0.657,	0.659,
cD3	1.170,	1.080,	0.835,	0.793,	0.924,	1.024,	1.206,	0.733,	1.101,	1.257,	0.884,	0.966,
cD2	1.040,	1.615,	1.269,	1.265,	1.151,	1.319,	1.158,	0.944,	1.353,	1.333,	0.997,	1.115,
cD1	0.657,	1.428,	0.707,	0.860,	0.580,	0.718,	0.689,	0.508,	0.783,	2.714,	0.910,	1.046,

The distortion measured is maximum in cD3, cD2, cD1 bands, which are usually ignored as they contains high freq components.

So data retrieved from the images is quite good.

Split the data & Feed to the Model



Results

- Without 5 fold CV:-
  - Accuracy: 0.8780
  - Recall: 0.9286
  - Precision: 0.9029
  - AUC: 0.9426

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 625, 12)]         0
_________________________________________________________________
conv1d (Conv1D)              (None, 625, 128)          12416
_________________________________________________________________
batch_normalization (BatchNo (None, 625, 128)          512
_________________________________________________________________
activation (Activation)      (None, 625, 128)          0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 625, 256)          164096
_________________________________________________________________
batch_normalization_1 (Batch (None, 625, 256)          1024
_________________________________________________________________
activation_1 (Activation)    (None, 625, 256)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 625, 128)          98432
_________________________________________________________________
batch_normalization_2 (Batch (None, 625, 128)          512
_________________________________________________________________
activation_2 (Activation)    (None, 625, 128)          0
_________________________________________________________________
global_average_pooling1d (Gl (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 277,121
Trainable params: 276,097
Non-trainable params: 1,024
_________________________________________________________________
```

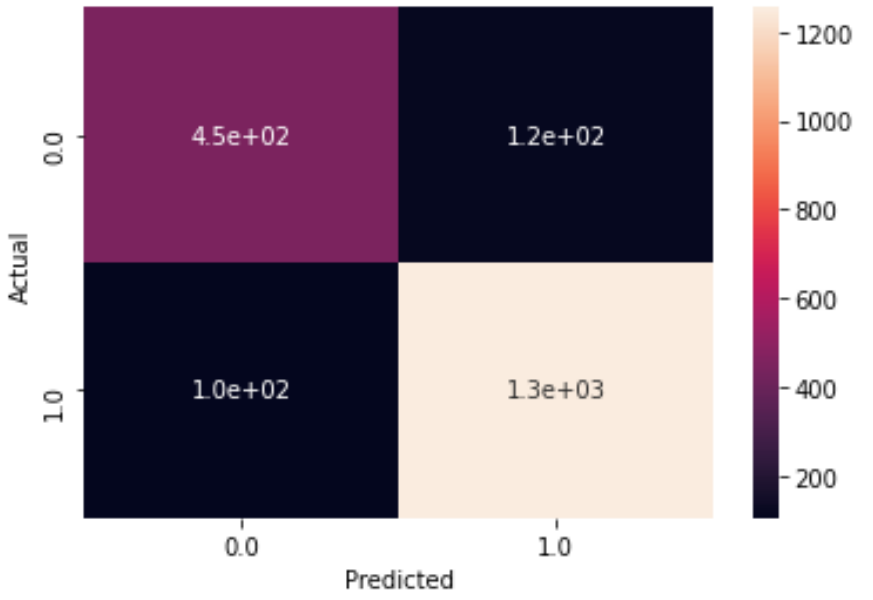
- After 5 fold Cross Validation:-

  (created 5 models and take average of the predicted output)

  - Accuracy: 0.88

Report & Confusion Matrix

	precision	recall	f1-score	support
HC	0.79	0.81	0.80	555
MI	0.92	0.91	0.92	1372
accuracy			0.88	1927
macro avg	0.86	0.86	0.86	1927
weighted avg	0.89	0.88	0.89	1927



Comparison with SOTA paper[2]

Classifiers	Sensitivity(%)	Accuracy(%)	Specificity(%)
KNN	85	81	77
SVM Lin	90.42	89	87.69
SVM RBF	93	96	99
KNN*	88.23	83.07	69.81
SVM Lin*	81.25	81.73	84.56
SVM RBF*	93.53	91.31	85.77
DNN Model	92.86	87.80	78.94

* (Implemented SOTA paper Result)

[2] Multiscale Energy and Eigenspace Approach to Detection and Localization of Myocardial Infarction https://ieeexplore.ieee.org/document/7047810

Future Work

- Found a new dataset PTB-XL[1] published on Apr 2020.
- Features:-

  Dataset of 21837 clinical 12-lead ECGs from 18885 patients of 10 second length.

  Sampling Freq: 100 Hz & 500 Hz

#Records	Superclass	Description
9528	NORM	Normal ECG
5486	MI	Myocardial Infarction
5250	STTC	ST/T Change
4907	CD	Conduction Disturbance
2655	HYP	Hypertrophy

[1] https://www.physionet.org/content/ptb-xl/1.0.1/

- Needed some pre-processing work to segregate the data into different classes.

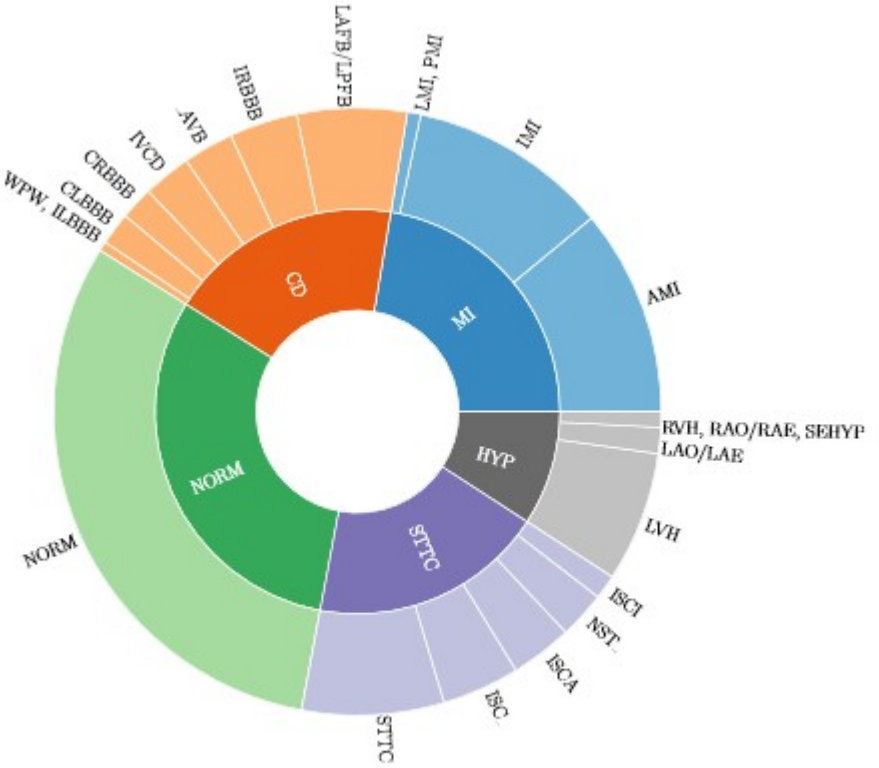


Figure 1: Graphical summary of the *PTB-XL* dataset in terms of diagnostic superclasses and subclasses, see Table 5 for a definition of the used acronyms.