Formula Derivation :

If we look at the orignal equation it represents as parabola where c intercept is 0, and each point on that parabola represents a Hr value , x axis represent corresponding Ha*Hb value
For deriving the formula initially we tried to come up with a formula using which we can Calculate the hydropathy value without having to shift the scale, to do this we assumed that there are 3 max points in hr values which get at minimum Ha* Hb vlaue, mid Ha*Hb value and Max Ha*Hb for which there are 3 Ha*Hb values, so assuming we have X1, X2, X3 , min , mid and max values of Ha*Hb respectively, we can get using the scales,
So Our equation becomes

-a X1^2 + b X1 + c = 0 -> min
-aX2^2+ b X2 + c = 1 -> mid
-aX3^2 + b X2 + c =0 -> max

now we have to get this X1, X2 , X3, from the list Ha*Hb value which we calculated, but the Problem with approach is that there is no guarantee that we identify which Ha*Hb is the mid value because the distribution of , we can however find min and max value accurately, but then the above equation fails as we will only have 2 equation and 3 unknown variables

Part 2 of the deriving the equation, here we decide to shift the scales such that all the values of Ha*Hb on the right hand side of the origin and on x axis, by doing this we make sure that c which represent the Y intercept in the equation is removed, now that all the values lie on the positive side or x axis and values of hr cannot be negative , we know the range of hr will only be from 0 to 1 , we know this from the orignal research paper.

So now equation becomes

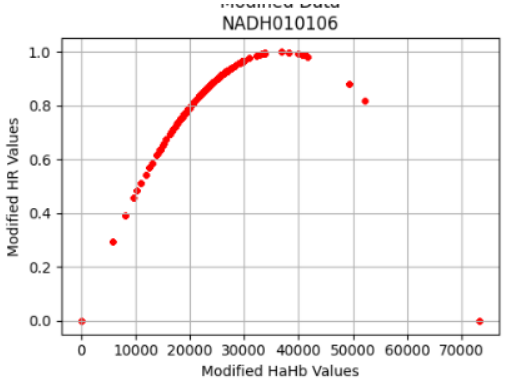
-aX^2 + b X1= 0
-aX2^2 + bX2 = 0

On solving the above equation we can represent a and b in the form of x1 and x2 , while getting the values of X1 and X2, we combine the both test and train data ,

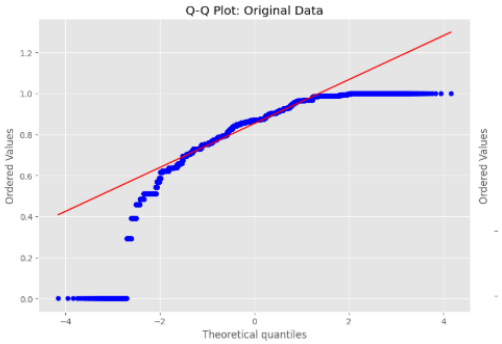
Data Analysis :

Now that we have the formula for generating hr values for all the scales, we have generated data we need to visualize the data,

Distribution of data for hr values for this we have ploted scatter plot , where x axis represents Ha*Hb value and y axis generated hr vlaue,



Straight away we can see the data is not evenly distributed , the above graph shows distribution of hr combined test and train is highly skewed to verify thi we can have plotted Q-Q plot, bellow is the qq plot which confirms that our data is left skewed.



We have performed this analysis on all the scales and observed almost all the generated values were either highly left or rightly skewed or were not evenly distributed,

- this can lead to Skewed inputs lead to slow or unstable training because activation and gradients become imbalanced in case of CNN.
- Also if we observed there are lot of gaps in between the data point, lets supposed we are training any model to identify patterns between 1 to 10 numbers but we had training data lieing between 1 to 3 and testing data between 2 to 5, we can see that model is well train and tested between the value of 1 to 5 but it is completely unaware of the data 5 to 10, this can be problematic if we are to deploy the model in the real world.
- Real world Issue of noise, we know the data collected for shape and electro, distance might have noise issue due, as it collected from electronic devices, as it is one of the common noise, however it can varry based on how the data has been collected
- In order to mimic the measurement uncertainty inherent in real-world and thereby improve the robustness of our predictive model, we augmented our simulated dataset by superimposing additive Gaussian noise on all input features. Specifically, following the approach of Suawa et al., we injected zero-mean Gaussian perturbations with standard deviations tuned to correspond to signal-to-noise ratios. This noise augmentation help not only approximates realistic experimental conditions, but also acts as a regularizer

during training—preventing overfitting to idealized data and promoting the learning of more generalizable feature representations.

Data Preprocessing

Based on the analysis of the data we decided to preprocess the data we decided to transform the training data first , in which tried various type of transformation log transformation , square root transformation , yeo jhonson and box-cox transformation,

Process followed during the transformation :

Step 1 :

Added Gaussian noise to the training data. We add this specifically to the trainging data .

Step 2 :

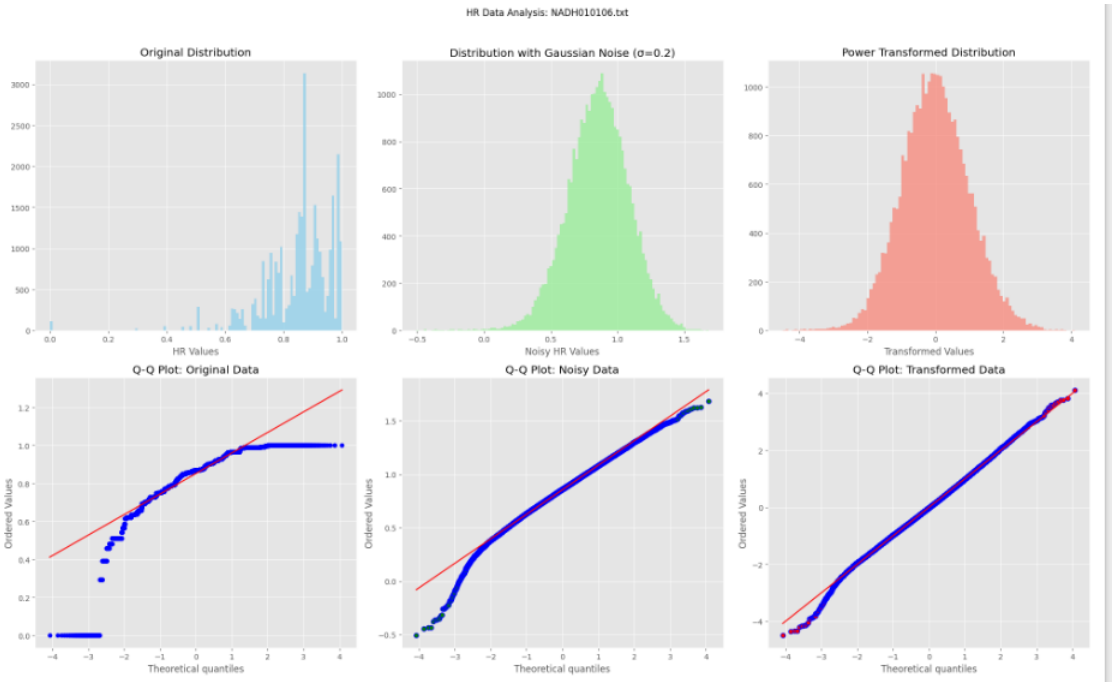
Apply the transformation to the training data, store the learning from training data later use this learning, and apply same transforming to the testing data or any other that is being given to model after the training

step 3 :

Select the best transformation based on the q-q Plot analysis

Results of Transformation and analysis :

After testing various transformation we found yeo - jhonson transformation to the best fit for our each in as it normalized the data effectively for each scale :
bellow is sample of transformation one of the data



Upon analyzing we can see that data is regularized after adding gausian noise, data is furthere normalized by the power transformation, but the problem is our data is no longer between 0 and 1 , range has change after applying yeo-jhonson transformation and after adding gaussian noise, so to bring it back to range of 0 and 1 we apply min max scaller.

Note : we take all the learning of yeo-jhonson and min max scaller, use to transform input data of test to maintain the consistancy