IBM MACHINE LEARNING: TIME SERIES PROJECT

Anomaly Detection using LSTM Autoencoders of NASA Bearing Dataset

1. Objective

Fault prognosis and diagnosis of Machines has become an important subject in the field of Mechanical Engineering. Early prediction and diagnosis can help industries to be more productive and economical. Prognosis and Diagnosis has become an integral part of Industry 4.0.

One of the Diagnosis methods is Anomaly Detection which is used in many fields like engineering, banking, finance etc. One such field where anomaly detection can be used is in the field of mechanical engineering. Anomaly detection can be used to diagnose the onset of failure in mechanical components. The failure can be categorised as an anomaly. In order to detect anomaly, we generally require sensor data like vibration data or force data or acoustic emission data or temperature data.

In this project, vibration data from bearing is used to detect anomalies. The bearing data is obtained from the NASA Bearing Dataset.

2. Data

The Data is obtained from NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA. The dataset contains 3 sets of Vibration data where in each set 4 bearings are tested till, they fail. In this project we make use of the Dataset number 2 which contains 984 data files.

Four bearings were installed on shaft and a motor was coupled to the shaft. A radial load of 2000 lbs were applied and run at constant speed of 2000 RPM. Rexnord ZA-2115 double row bearings were used. The accelerometer measures the vibrations during the tests, which is used for anomaly detection. All failures occurred after more than 100 million revolutions.

Three (3) data sets are included in the data packet (IMS-Rexnord Bearing Data.zip). Each data set describes a test-to-failure experiment. Each data set consists of individual files that are 1-second vibration signal snapshots recorded at specific intervals. Each file consists of 20,480 points with the sampling rate set at 20 kHz. In this project Dataset number 2 was used. The details of Set number 2 are given below.

- Recording Duration: February 12, 2004 10:32:39 to February 19, 2004 06:22:39
- No. of Files: 984
- No. of Channels: 4
- Channel Arrangement: Bearing 1 – Ch 1; Bearing2 – Ch 2; Bearing3 – Ch3; Bearing 4 – Ch 4.

- File Recording Interval: Every 10 minutes
- File Format: ASCII
- Description: At the end of the test-to-failure experiment, outer race failure occurred in bearing 1

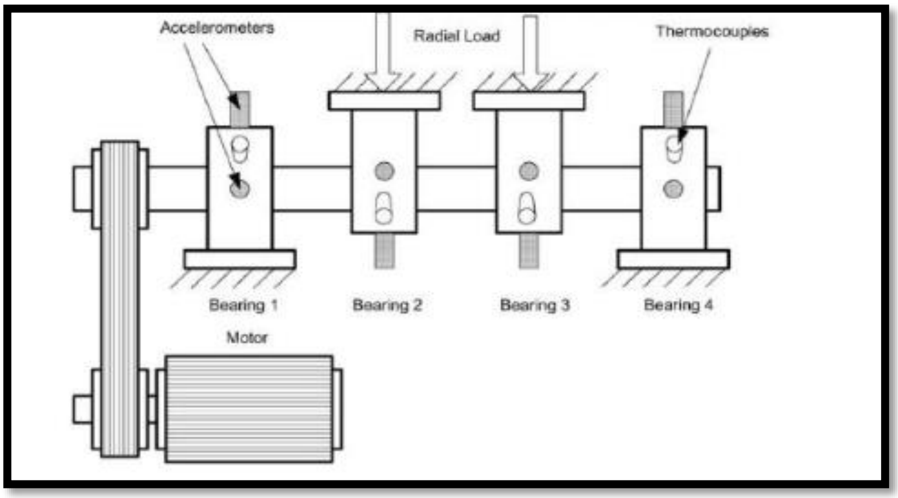

Figure 1. Test Setup

3. Data Cleaning/Pre-processing.

3.1 Dimensionality Reduction:

Since there are 984 files and each file contains more than 20,000 readings for each of the four bearings, we cannot feed the entire dataset to our autoencoder. Therefore, in order to reduce the dimensions of the data, we use the mean absolute value of the entire file. Thus, we have 984 reading for each of the four bearings making a total of 3936 readings.

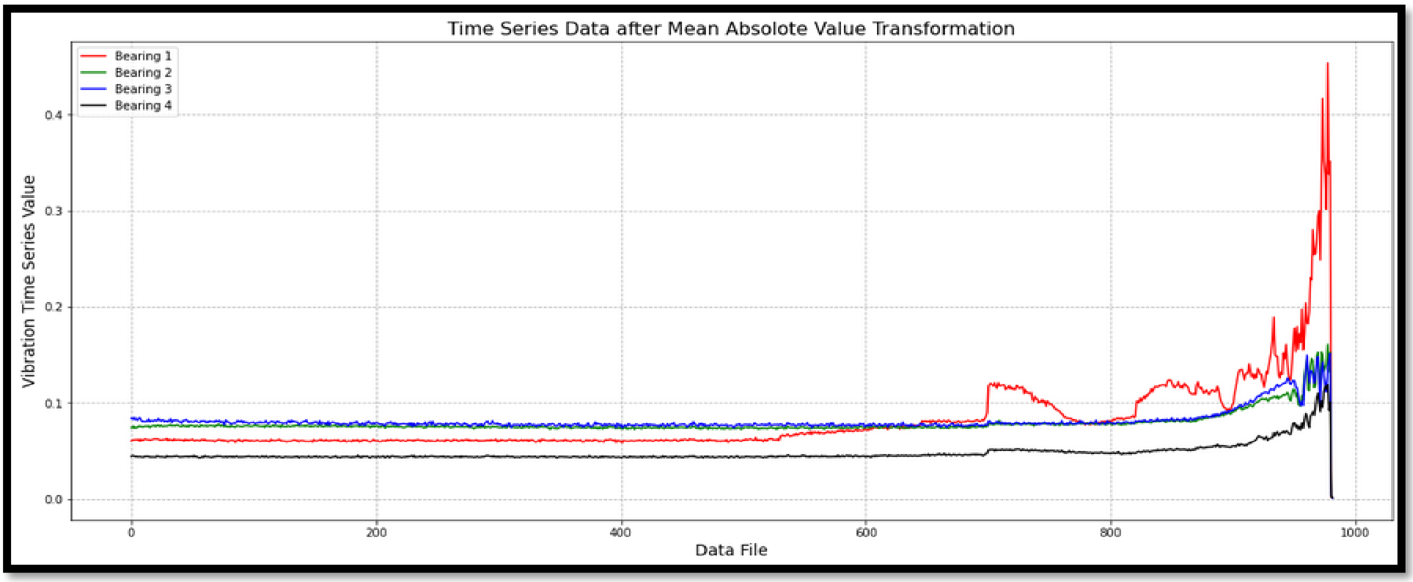

Figure 2. Dimensionality Reduction

The resulting time series is a univariate time series having an upwards trend toward the end of the series.

3.2 Formatting Data into LSTM format:

We use the data from Bearing 1 as the training set and the Bearing 2 the test set. Before passing the data to the autoencoder, we need to prepare it in the data form that LSTM acceptable form. LSTM expects data to be in the form of a 3-D array which has the following dimensions.

LSTM Format ➔[batch_size, time_steps, input_dimension]

Given below is the format of how our time series needs to be structured.

If our time series is ===>> [1,2,3,4,5,6,7,8,9]

- n_samples = 9
- n_sequence = 3 (Assume)

We should make our time series as:

Feature ===> Labels

[1,2,3] ===> [4]

[2,3,4] ===> [5]

[3,4,5] ===> [6]

[4,5,6] ===> [7]

[5,6,7] ===> [8]

[6,7,8] ===> [9]

3.3 Normalizing:

Both the train and test set are normalized to the range if -1 to 1. Normalizing is done with the help of the below given formula.

$$X_{normalized} = \frac{(\ x_i - \max(x)\ ) + (\ x_i - \min(x)\ )}{\max(x) - \min\ (x)}$$

Where, *x is the time series data.*

4. Methodology

We use LSTM in our auto encoder model. In order to get the best model, we use 3 different model with different hyperparameters. The architecture is given below.

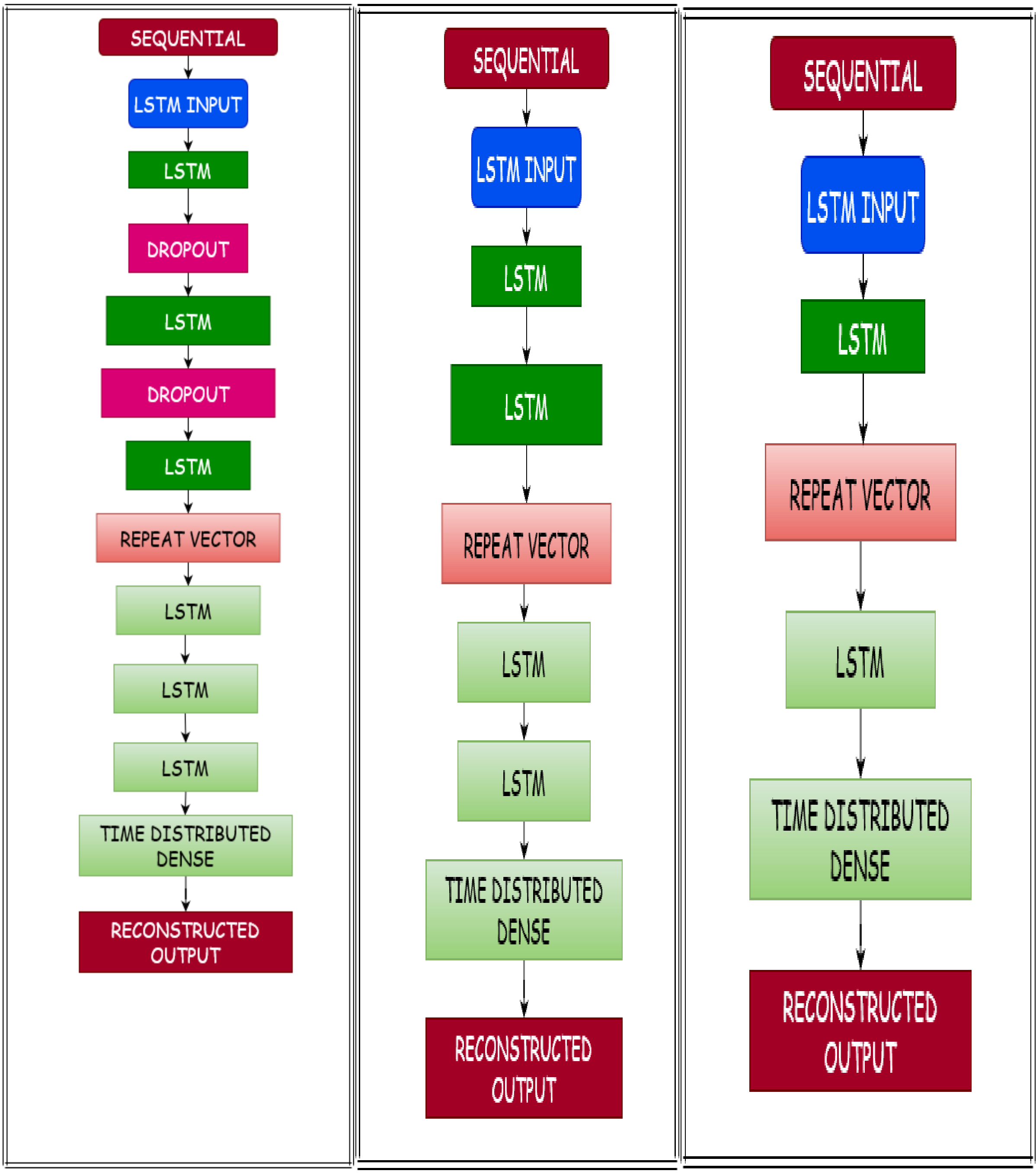

Figure 3. Architectures of Model 1, Model 2 Model 3 respectively from left to right

Model 1:

The model 1 contain 3 LSTM layers in the encoder layer and 3 more in the decoder layer. The Number of units in the 3 layers are 64, 32 and 16 respectively. There are two dropout layers to avoid overfitting with a dropout rate of 0.2. Return sequences is set true in the first two LSTM layers in the encoder zone.

In the decoder layer, the number of units are 16, 32 and 64 respectively with return sequences set to true in all layers. The last layer is the reconstruction of the original sequence layer. It is a time distributed dense layer with number of units equal to the input dimension of the passed 3-D Array. All the layers have 'relu' activation.

```
Model: "LSTM_Autoencoder_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM_1 (LSTM)                (None, 30, 64)            16896
_________________________________________________________________
Dropout_1 (Dropout)          (None, 30, 64)            0
_________________________________________________________________
LSTM_2 (LSTM)                (None, 30, 32)            12416
_________________________________________________________________
Dropout_2 (Dropout)          (None, 30, 32)            0
_________________________________________________________________
LSTM_3 (LSTM)                (None, 16)                3136
_________________________________________________________________
Repeat_Vector (RepeatVector) (None, 30, 16)            0
_________________________________________________________________
LSTM_4 (LSTM)                (None, 30, 16)            2112
_________________________________________________________________
LSTM_5 (LSTM)                (None, 30, 32)            6272
_________________________________________________________________
LSTM_6 (LSTM)                (None, 30, 64)            24832
_________________________________________________________________
time_distributed (TimeDistri (None, 30, 1)             65
=================================================================
Total params: 65,729
Trainable params: 65,729
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. Model 1 Summary

Model 2:

The model 2 contain 2LSTM layers in the encoder layer and 2 more in the decoder layer. The Number of units in the 2 layers are 32 and 16 respectively. Return sequences is set true in the first LSTM layers in the encoder zone.

In the decoder layer, the number of units are 16 and 32 respectively with return sequences set to true in all layers. The last layer is the reconstruction of the original sequence layer. It is a time distributed dense layer with number of units equal to the input dimension of the passed 3-D Array. All the layers have ‘relu’ activation.

```
Model: "LSTM_Autoencoder_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM_1 (LSTM)                (None, 30, 32)            4352
_________________________________________________________________
LSTM_2 (LSTM)                (None, 16)                3136
_________________________________________________________________
Repeat_Vector (RepeatVector) (None, 30, 16)            0
_________________________________________________________________
LSTM_3 (LSTM)                (None, 30, 16)            2112
_________________________________________________________________
LSTM_4 (LSTM)                (None, 30, 32)            6272
_________________________________________________________________
time_distributed_1 (TimeDist (None, 30, 1)             33
=================================================================
Total params: 15,905
Trainable params: 15,905
Non-trainable params: 0
_________________________________________________________________
```

Figure 5. Model 2 Summary

Model 3:

The model 3 contain 1 LSTM layers in the encoder layer and 1 more in the decoder layer. The Number of units are 16. Return sequences is set to false in the first LSTM layers in the encoder zone.

In the decoder layer, the number of units are 16, with return sequences set to. The last layer is the reconstruction of the original sequence layer. It is a time distributed dense layer with number of units equal to the input dimension of the passed 3-D Array. All the layers have ‘relu’ activation.

```
Model: "LSTM_Autoencoder_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM_1 (LSTM)                (None, 16)                1152
_________________________________________________________________
Repeat_Vector (RepeatVector) (None, 30, 16)            0
_________________________________________________________________
LSTM_2 (LSTM)                (None, 30, 16)            2112
_________________________________________________________________
time_distributed_2 (TimeDist (None, 30, 1)             17
=================================================================
Total params: 3,281
Trainable params: 3,281
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. Model 3 Summary.

All Models are compiled with mean absolute error as the loss function and optimizer as Adam Optimizer.

5. Training and Results

5.1 Training

All the models were trained for 20 epochs with a 10 % data used as validation data. The losses on both data set reduced with model 3 having smooth loss curve. Given below are the loss evolutions at different epochs.

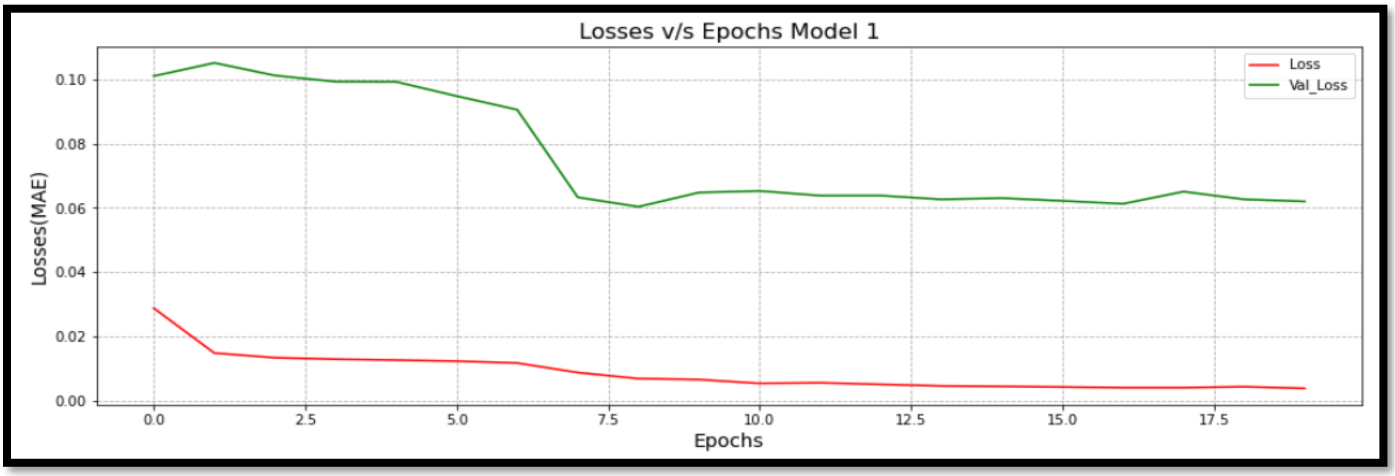

Figure 7. Model 1 Training Loss Evolution

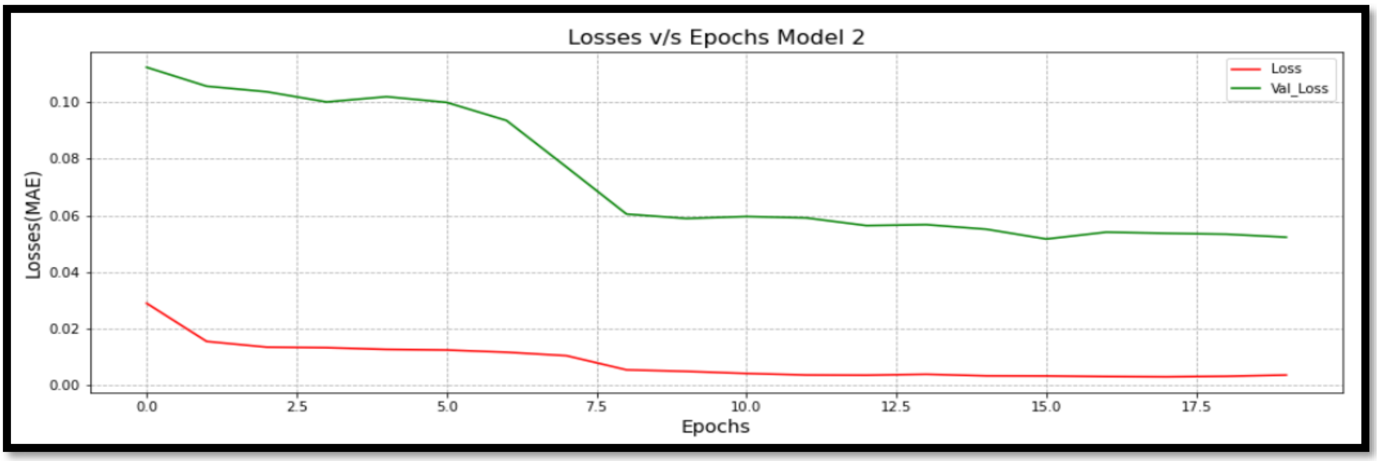

Figure 8. Model 2 Training Evolution.

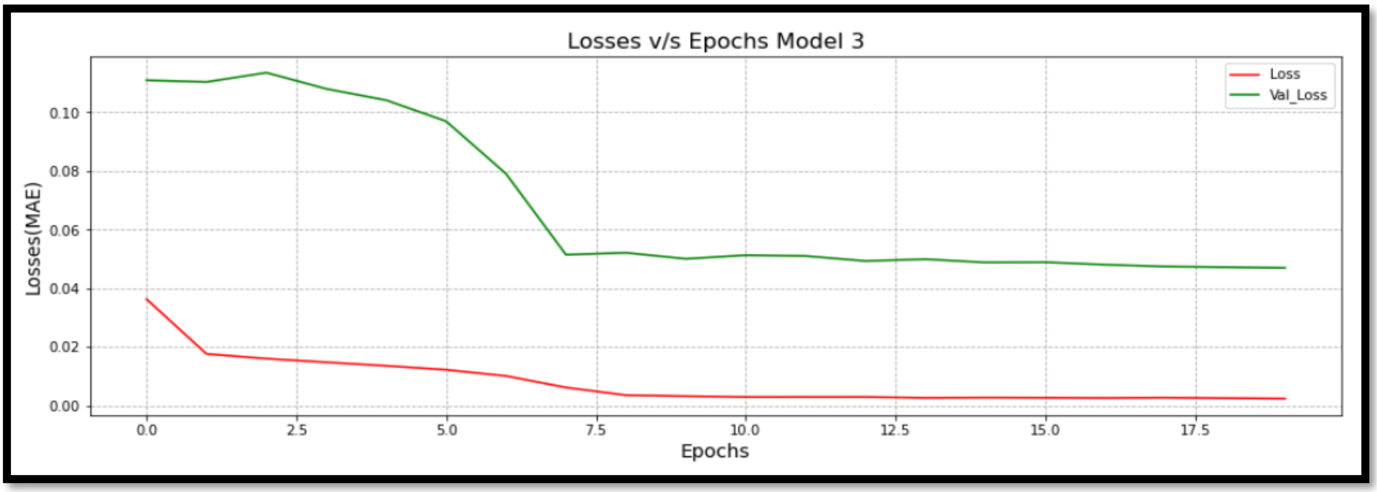

Figure 9. Model 3 Training Evolution.

5.2 Evaluation on Test Set:

All the models are tested on the test to see how it performs on unseen data. Given below are the results

Model Number	Loss
1	0.0359
2	0.05208
3	0.07049

Model 1 performed better than the other two on unseen data.

5.3 Anomaly Detection

In order to detect the anomaly, we plot the loss distribution histogram. The loss is the MAE between the predicted value by the autoencoder and the real value. In the histogram, we see a major chunk of the distribution towards the left of the plot. The distributions in the right of the plot with big MAE are decided as anomalies. By selecting a threshold, we tell if the MAE is above the threshold, it is an anomaly.

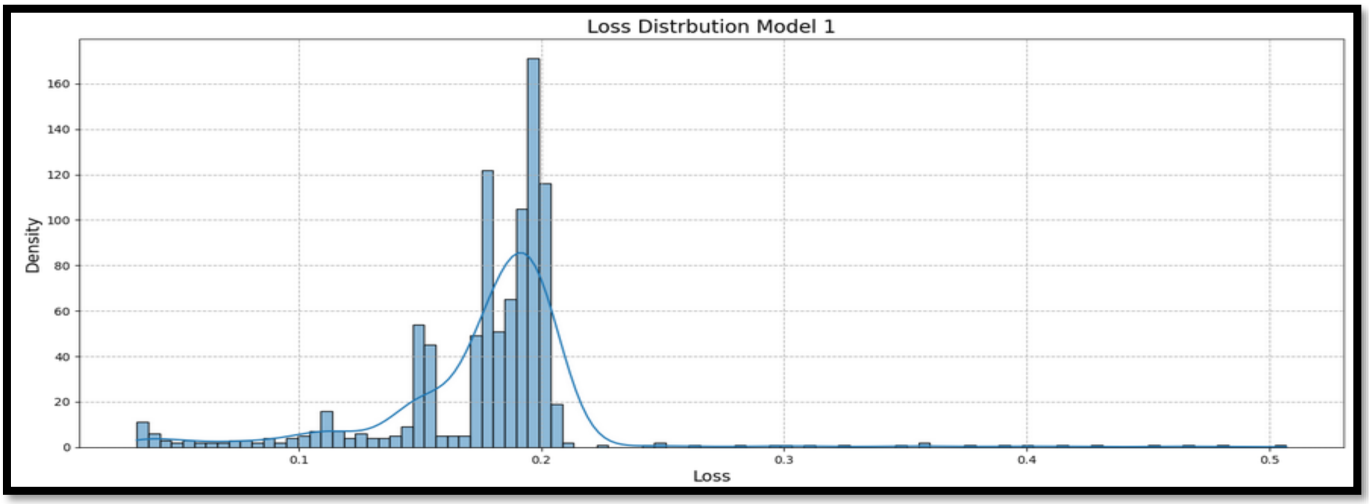

Figure 10. Model 1 Distribution of Loss.

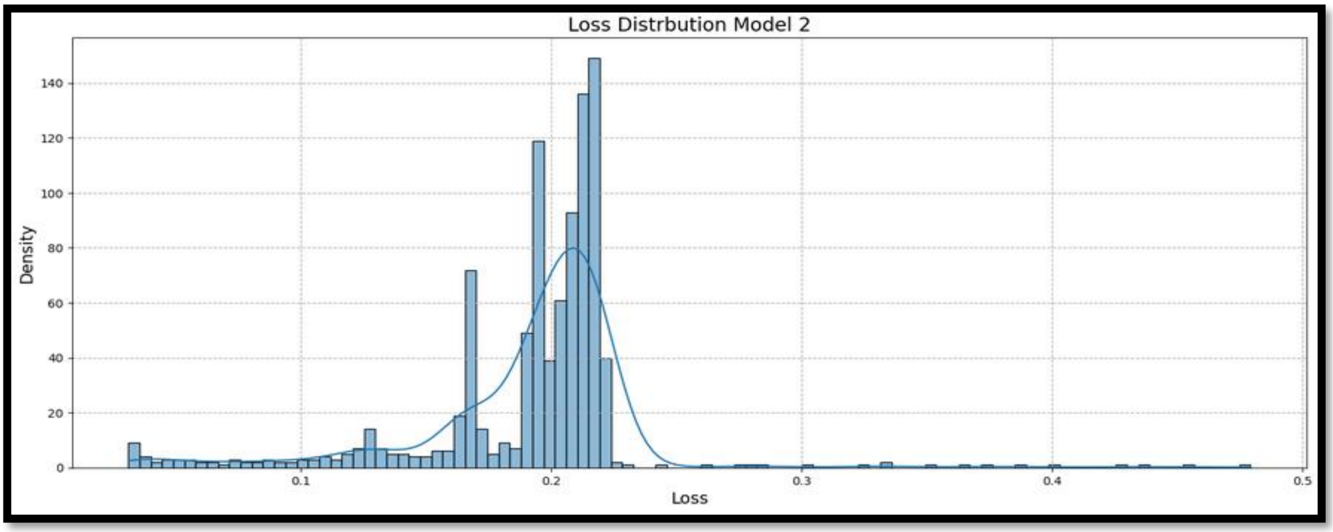

Figure 11. Model 2 Distribution of Loss.

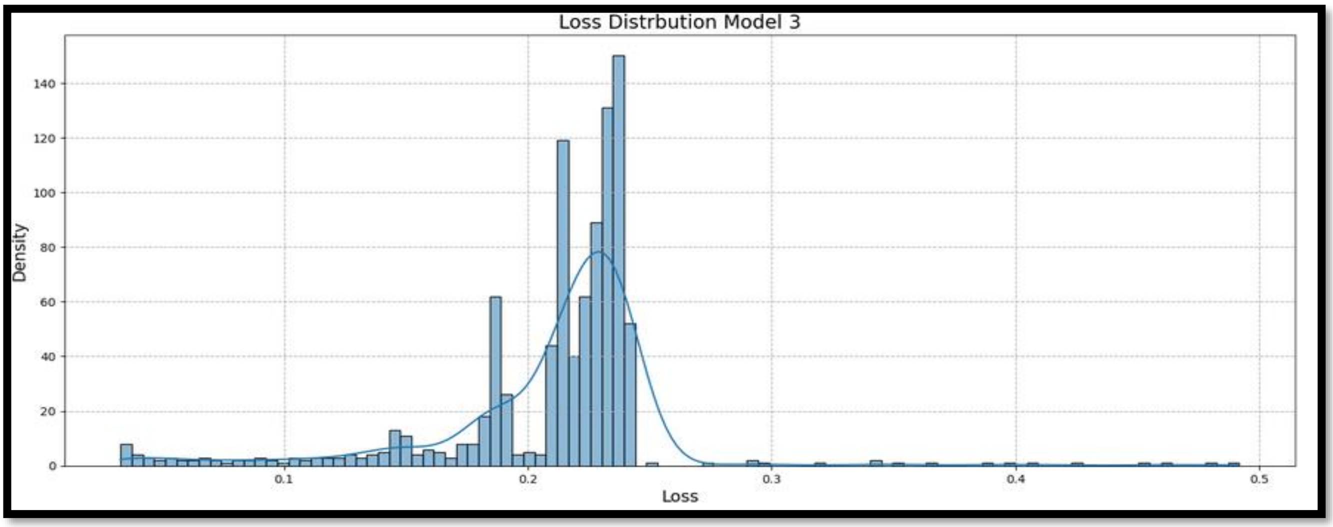

Figure 12. Model 3 Distribution of Loss.

From the above graphs the threshold is selected as follows:

1. Model 1: Threshold of 0.21
2. Model 2: Threshold of 0.24
3. Model 3: Threshold of 0.27

We describe anomaly as anything above the threshold loss as anomaly. In order to describe it better following are the points which the respective models describe as anomaly.

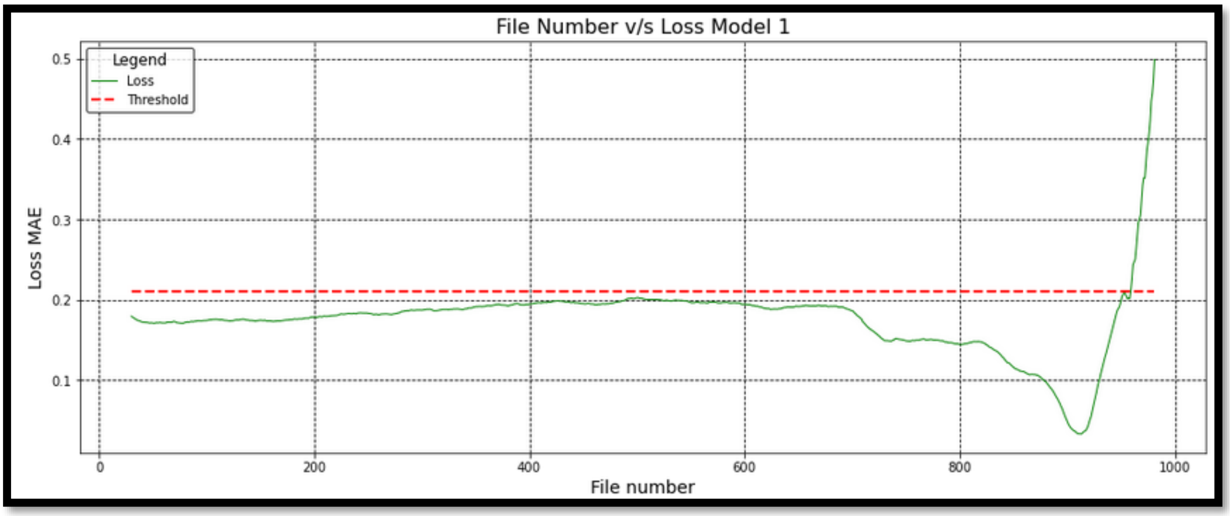

Figure 13. Model 1 Threshold.

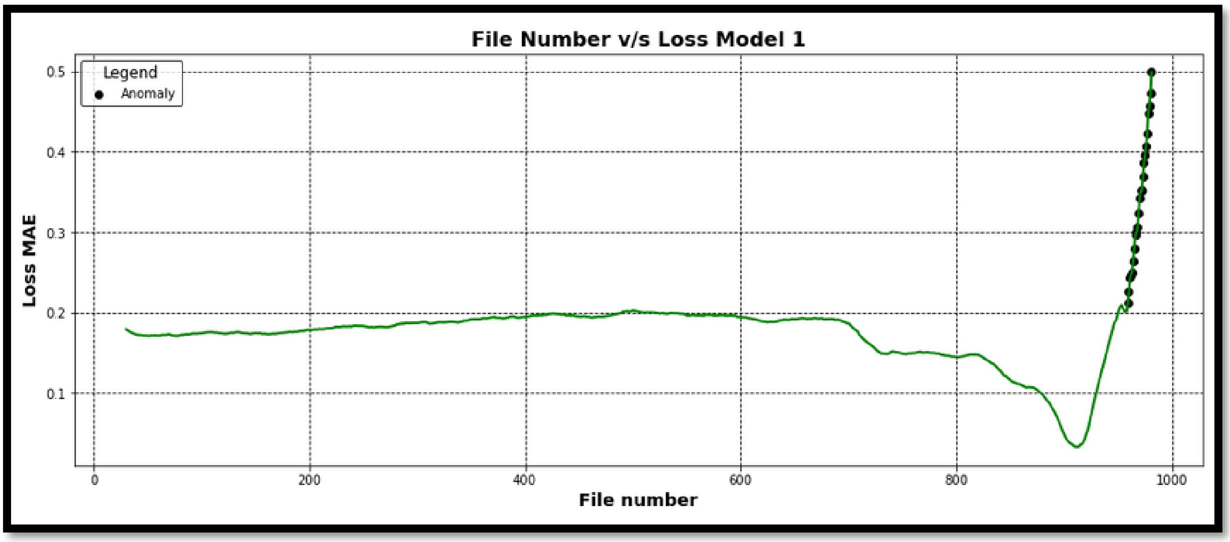

Figure 14. Model 1 Anomaly Points in black.

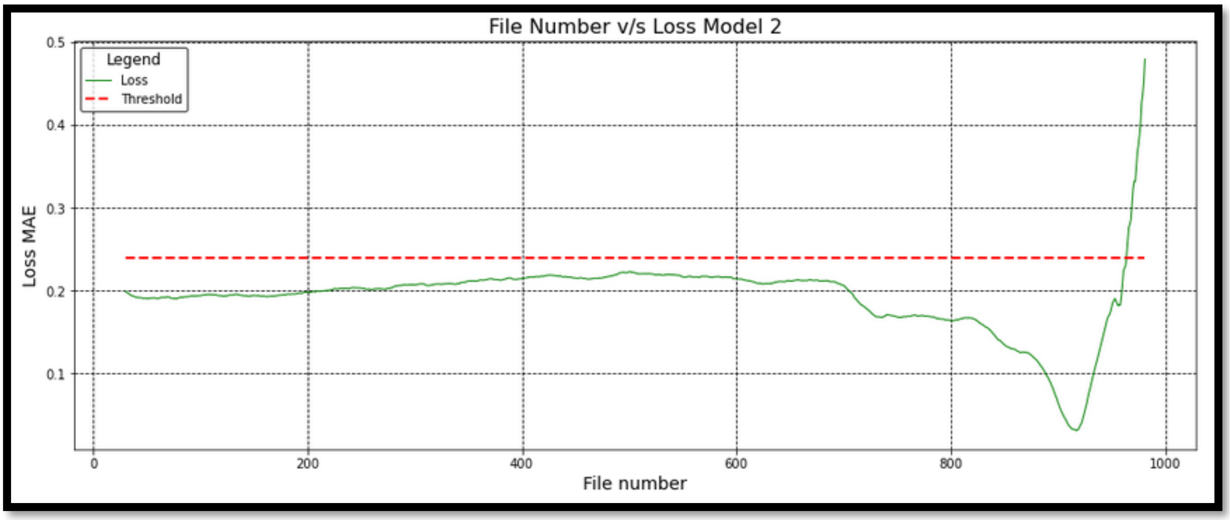

Figure 15. Model 2 Threshold Loss.

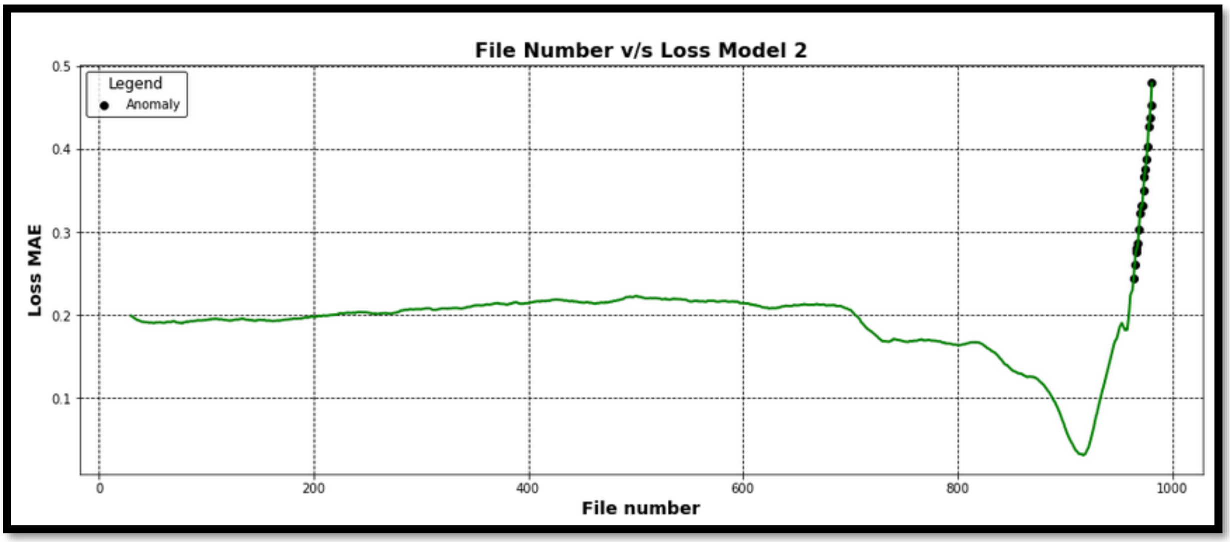

Figure 16. Model 2 Anomaly Points in black.

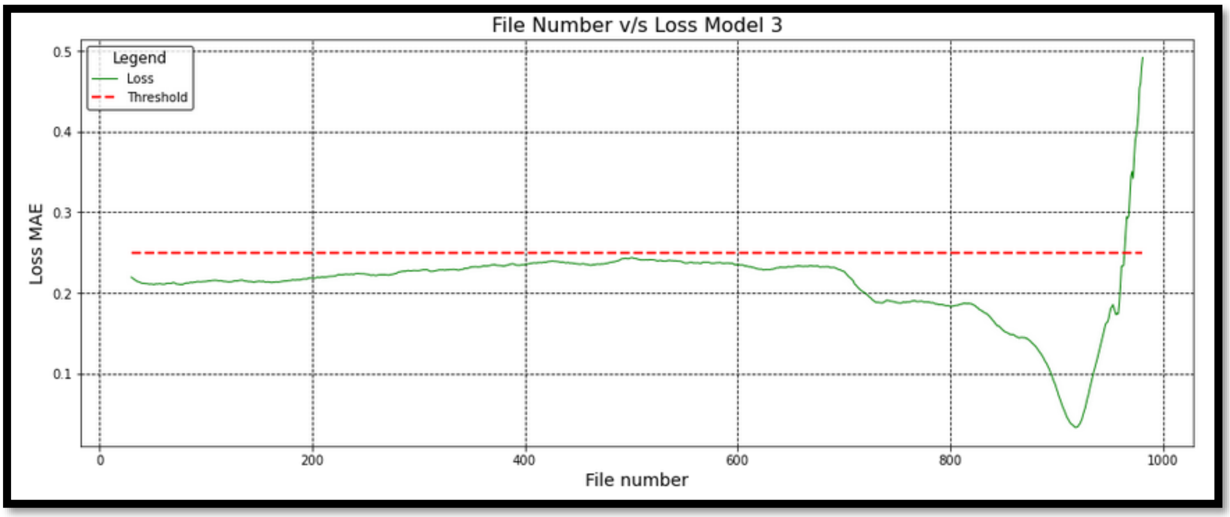

Figure 17. Model 3 Threshold Loss

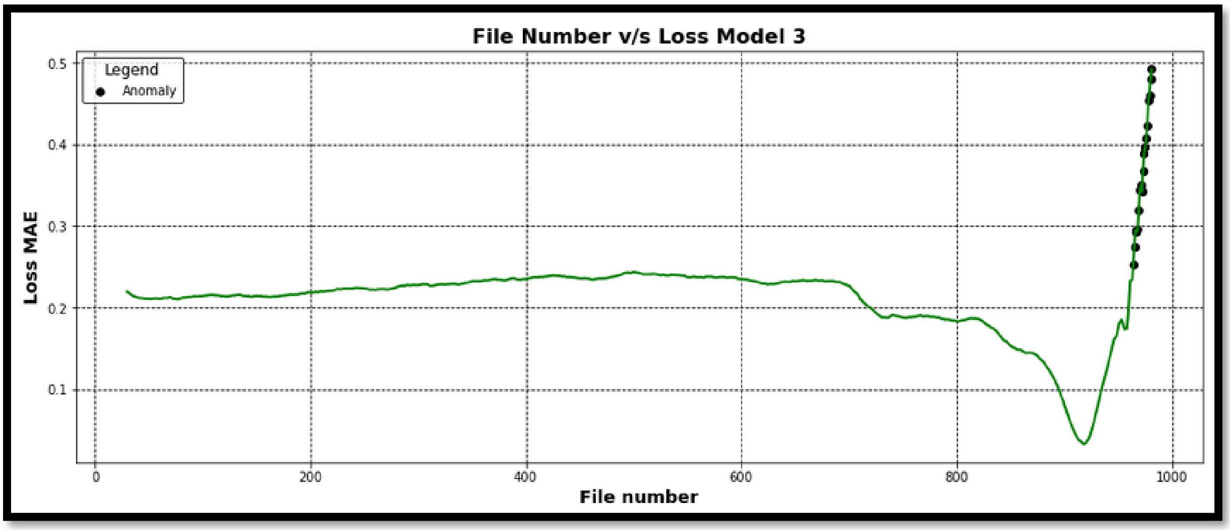

Figure 18. Model 3 Anomaly Points in black

6. Conclusion

From the results obtained, even though all the models performed very well, we can conclude that the Model 1 has the best performance based on the performance on the test set. Model 1 predicts more anomaly points than the other 2 models. The anomaly points can be concluded as bearing failure.

Further improvement on the model can be done by using Bearing 3 and Bearing 4 data. Also, we used only the mean absolute value of nearly 20000 data points and made it into a single data. This is a very crude way to crush data as we lose lot of information. Better dimensionality methods like piecewise aggregate approximation can be utilised to reduce dimensions.

The model can be further enhanced by using the NASA Bearing dataset number 1 and 3 which has more than 2000 and 4000 datafiles respectively. This can further generalise the model.

7. Reference

1. Data Reference:

- https://www.kaggle.com/rkuo2000/nasa-bearing-sensor-data
- https://www.kaggle.com/vinayak123tyagi/bearing-dataset
- http://ti.arc.nasa.gov/project/prognostic-data-repository

2. Reference Paper:

Hai Qiu, Jay Lee, Jing Lin. “Wavelet Filter-based Weak Signature Detection Method and its Application on Roller Bearing Prognostics.” Journal of Sound and Vibration 289 (2006) 1066-1090