

## KBO Services Test Report

Name: Shengbin Wang      Email: sambin\_wang@yahoo.com

The detailed explanation about each of method can be find in the corresponding code files.

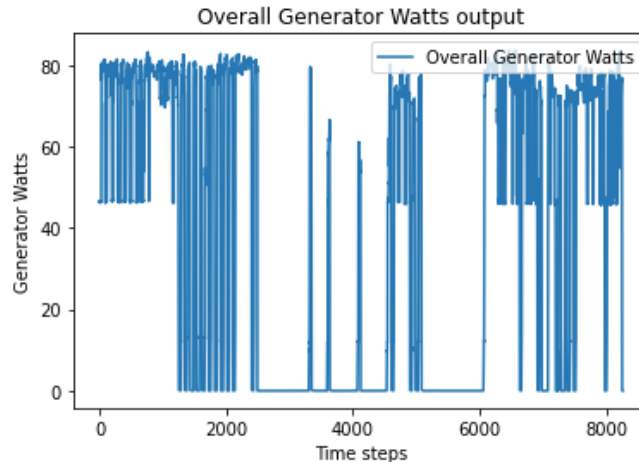
### Exploratory data analysis:

**a. There are five independent Variables and one dependent variable:**

- # Five independent Variables
  - 'GT1 Exhaust Temp Median Corrected By Average {Avg}'
  - 'GT1 Compressor Inlet Temperature {Avg}'
  - 'GT1 IGV angle in deg {Avg}'
  - 'GT1 Compressor Discharge Press Max Select {Avg}'
  - 'GT1 Compressor Discharge Temperature {Avg}'
- # One dependent Variable
  - 'GT1 Generator Watts Max Selected {Avg}'

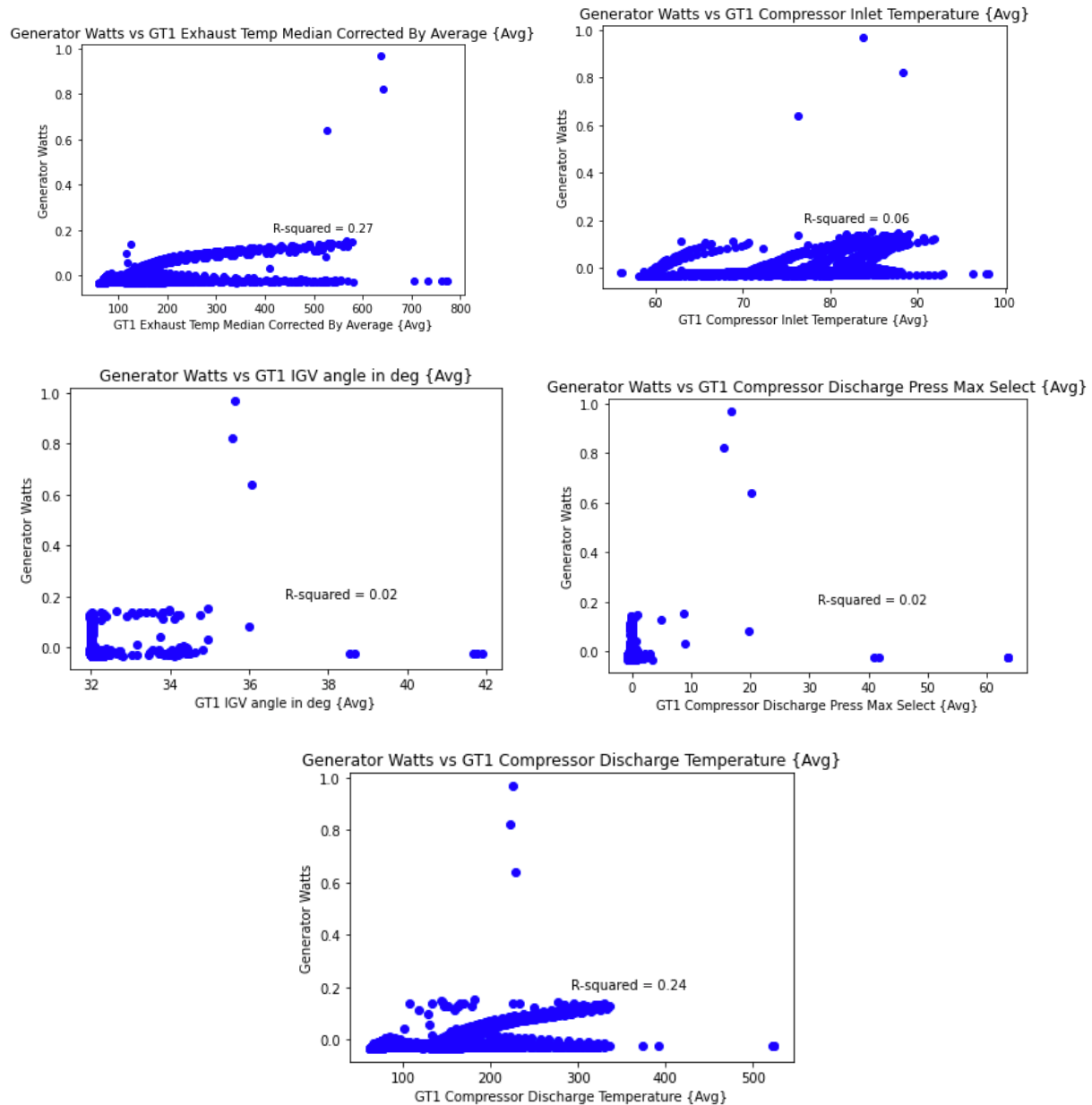
**b. Total 8260 records.**

**c. Generator Watts Max Selected {Avg} vs Time stamp:**



**Explanation:** There are some results = 0: gas turbine may be broken, or under maintenance, find the output below than 1, and the related variables.

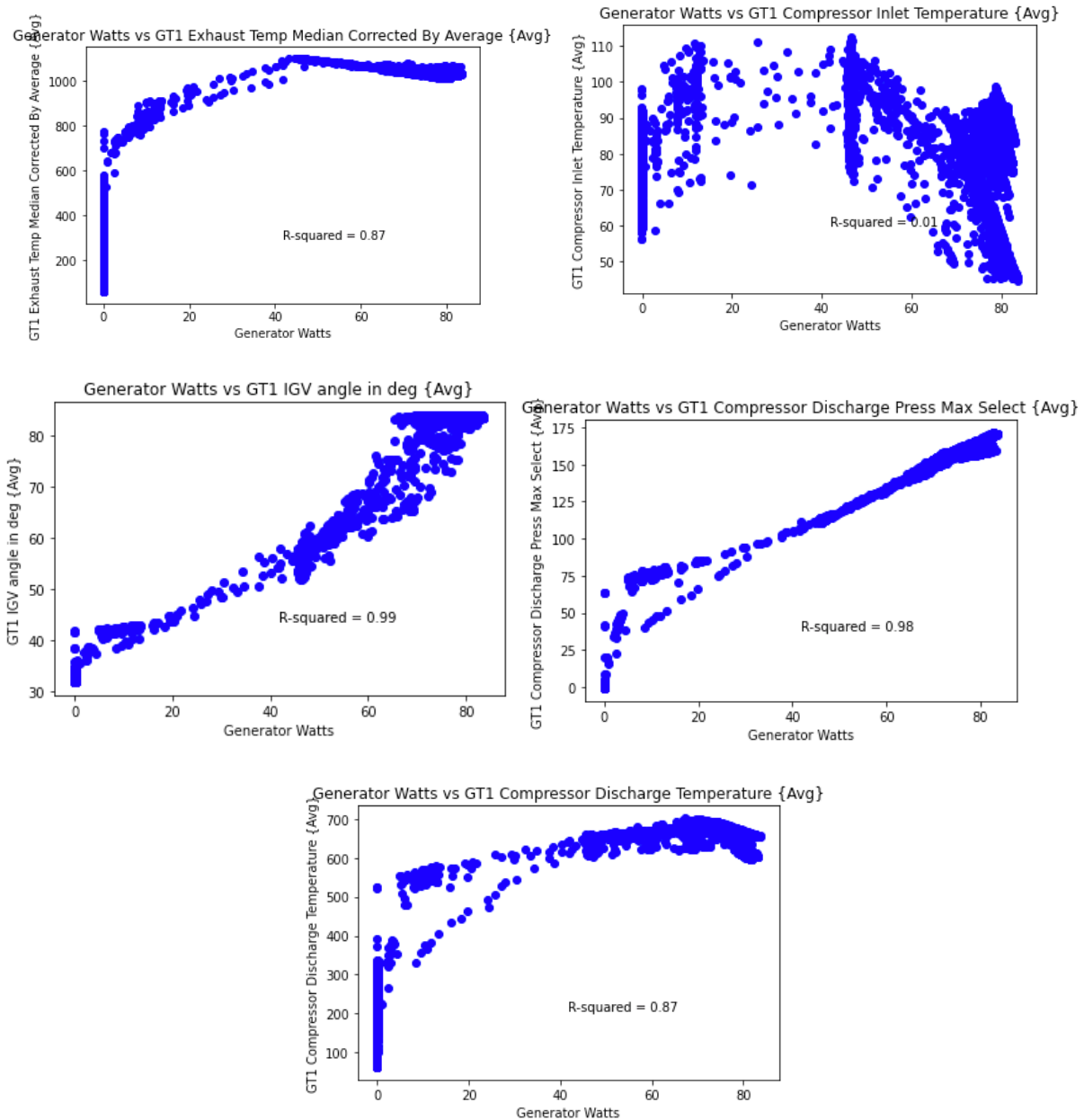
d. There are total 3754 records. Plot the Generator Watts vs variables



### Explanation:

- For variable: 'GT1 Compressor Discharge Press Max Select {Avg}'
  - most of the experimental data is concentrated in the lower left corner (0-5)
- For variable: "GT1 IGv angle in deg {Avg}"
  - most of the experimental data is concentrated in the lower left corner (32-35)
- Use these two variable to predict the output is 0. (broken, or under maintenance)

- e. By analyzing Pearson correlation between each of variable, and R-square between generates watts vs variable, I dropped variable: 'GT1 Compressor Inlet Temperature {Avg}', which has the lowest value. Plots shown below:



Below parts I will introduce four machine learning methods for this test.

## Method one: ANN

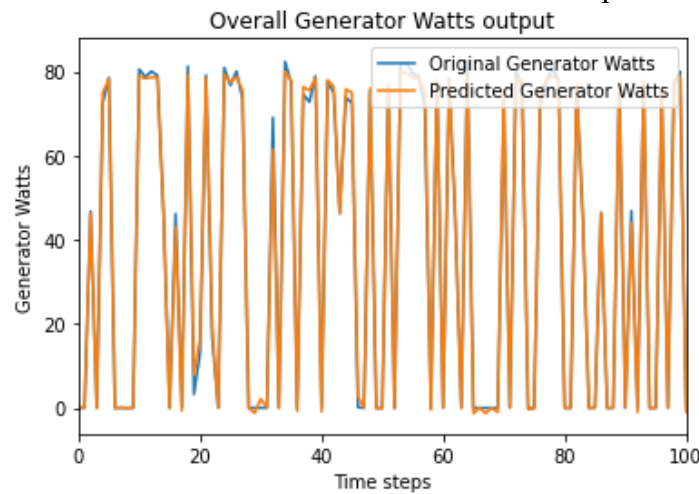
Code explanation: [https://github.com/ShengbinWang/KBO/blob/main/ANNturbine\\_noadd.ipynb](https://github.com/ShengbinWang/KBO/blob/main/ANNturbine_noadd.ipynb)

### 1. Finding best hyperparameters of ANN using GridSearchCV

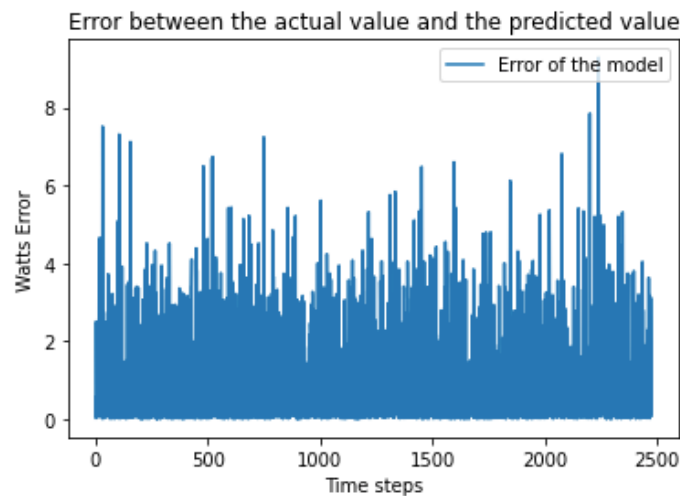
- Best parameters: {'Optimizer\_trial': 'adam', 'batch\_size': 10, 'epochs': 10}
- The  $R^2$  of ANN model is: 0.9979149809002641

### 2. Results show below:

- The predict value with the actual value within 100 timestamps:



- The Average Error: 1.088474810201012, the error for all of the test dataset shown below:



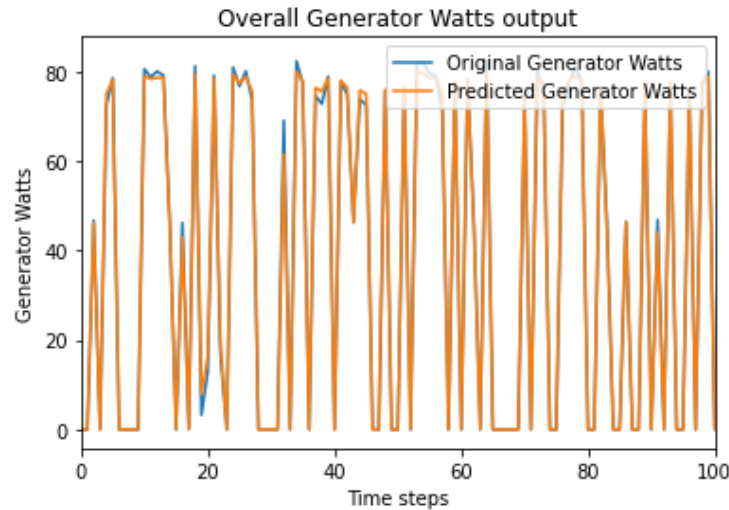
### 3. With the drop 0 method, when

- For variable: 'GT1 Compressor Discharge Press Max Select {Avg}' below 5
- For variable: "GT1 IGV angle in deg {Avg}" below 35

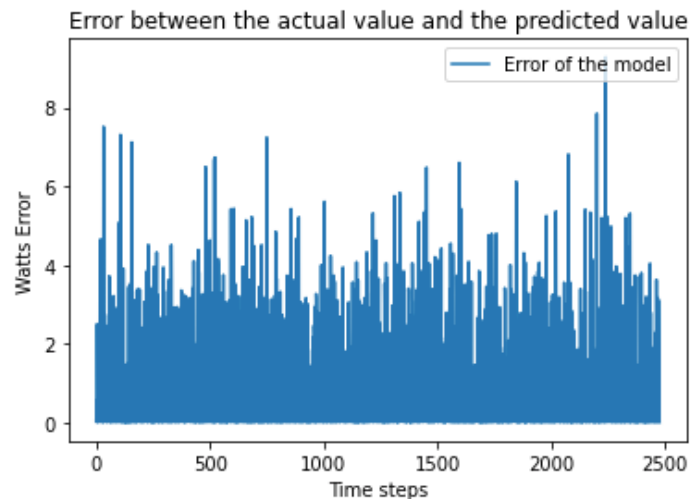
- The  $R^2$  of ANN model is: 0.9980423194726553,

#### 4. Results show below:

- The predict value with the actual value within 100 timestamps:



- The Average Error: 0.9463337103020106, the error for all of the test dataset shown below:



#### Reference:

- <https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/>
- <https://www.askpython.com/python/coefficient-of-determination>
- <https://www.bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-1-Introduction.php>
- <https://thinkingneuron.com/how-to-find-best-hyperparameters-using-gridsearchcv-in-python/>
- <https://thinkingneuron.com/using-artificial-neural-networks-for-regression-in-python/>

## ***Method two: ANN with new variable added.***

Code explanation: <https://github.com/ShengbinWang/KBO/blob/main/ANNturbine.ipynb>

### **1. Create more variable based on the variables provided:**

- Difference between two consecutive numbers, usually, sudden changes may lead to failure accident, decrease the Productivity. Because the value change for each of the variables will certainly affect gas turbine performance, like Compressor Inlet temperature suddenly change a lot.
- Average between three consecutive numbers, two will overlap. Avoid some sudden changes that lead to inaccurate data.
- Drop the NaN value, the final shape of the dataset: (8258, 16)

### **2. By analyzing Pearson correlation between each of variable, and R-square between generates watts vs variable. Selected variable based on above analyze:**

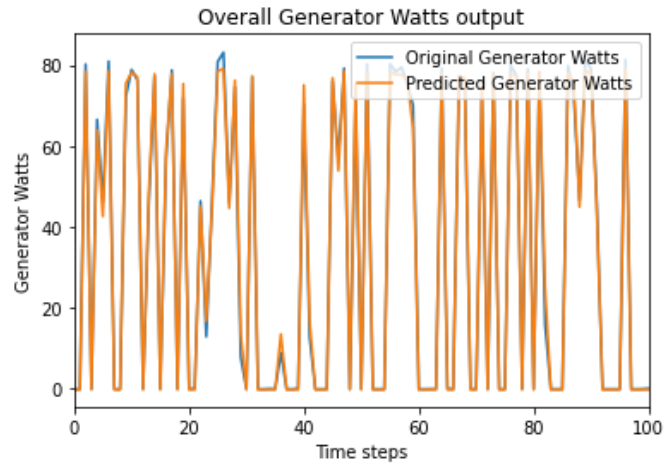
• # GT1 Exhaust Temp Median Corrected By Average {Avg}	0.934648
• # GT1 IGV angle in deg {Avg}	0.993491
• # GT1 Compressor Discharge Press Max Select {Avg}	0.990797
• # GT1 Compressor Discharge Temperature {Avg}	0.934783
• # med_exh_tem_average	0.934844
• # IGV_ang_average	0.989188
• # com_disch_pre_max_average	0.987999
• # com_disch_tem_average	0.935702

### **3. Finding best hyperparameters of ANN using GridSearchCV**

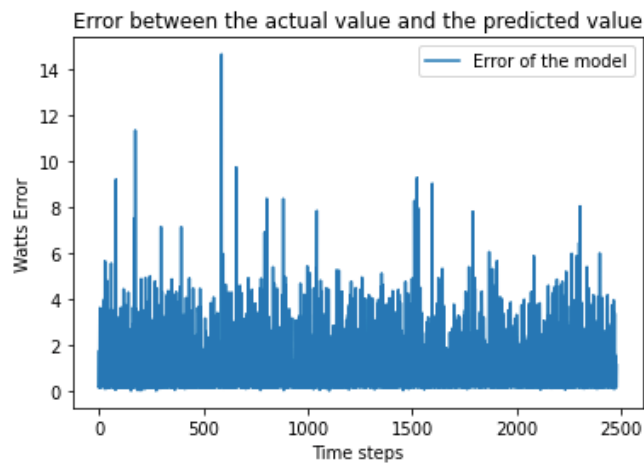
- Best parameters: {'Optimizer\_trial': 'adam', 'batch\_size': 10, 'epochs': 10}
- The  $R^2$  of ANN model is: 0.9989965497735229

### **4. Results show below:**

- The predict value with the actual value within 100 timestamps:



- The Average Error: 1.2249424420916317, the error for all of the test dataset shown below:

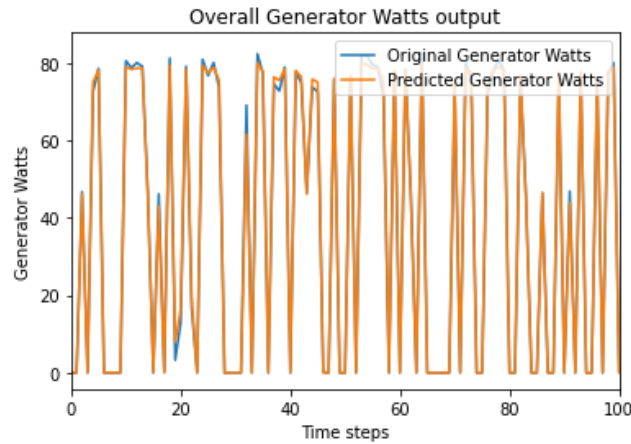


## 5. With the drop 0 method, when

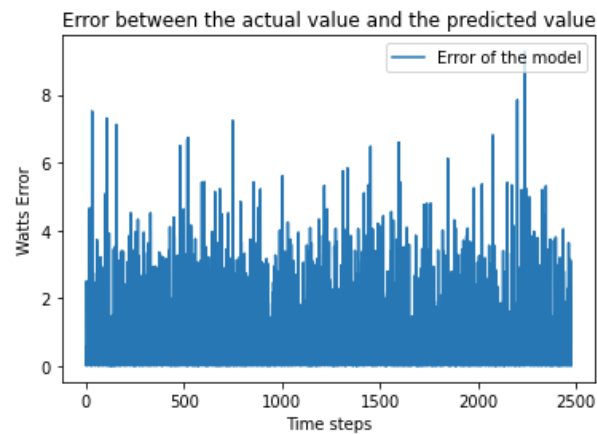
- For variable: 'GT1 Compressor Discharge Press Max Select {Avg}' below 5
- For variable: "GT1 IGV angle in deg {Avg}" below 35
- The  $R^2$  of ANN model is: 0.998993271027454

## 6. Results show below:

- The predict value with the actual value within 100 timestamps:



- The Average Error: 0.6589024247075902, the error for all of the test dataset shown below:



## Reference:

- <https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/>
- <https://www.askpython.com/python/coefficient-of-determination>
- <https://www.bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-1-Introduction.php>
- <https://thinkingneuron.com/how-to-find-best-hyperparameters-using-gridsearchcv-in-python/>
- <https://thinkingneuron.com/using-artificial-neural-networks-for-regression-in-python/>

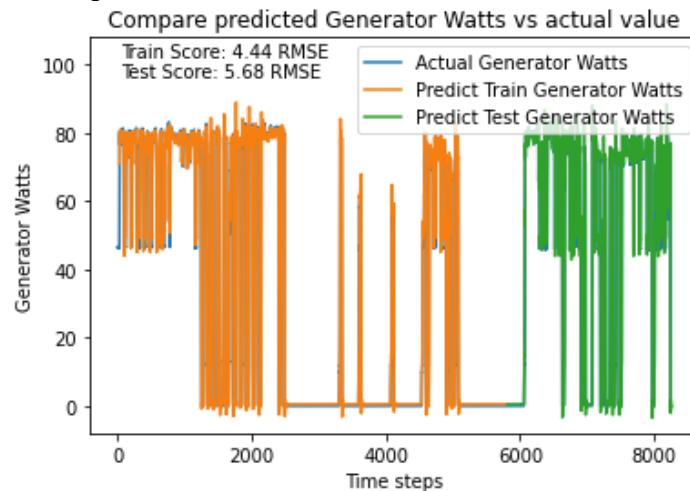


***Method three: LSTM with 30 time stamps look back.***

(The detailed explanation about this method can be find in the corresponding code file)

Code explanation: [https://github.com/ShengbinWang/KBO/blob/main/Turbine\\_LSTM.ipynb](https://github.com/ShengbinWang/KBO/blob/main/Turbine_LSTM.ipynb)

- The predict value with the actual value within whole timestamps, and the Train and test scores shown in the picture below:



**Reference:**

- <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- <https://stackabuse.com/time-series-analysis-with-lstm-using-pythons-keras-library/>

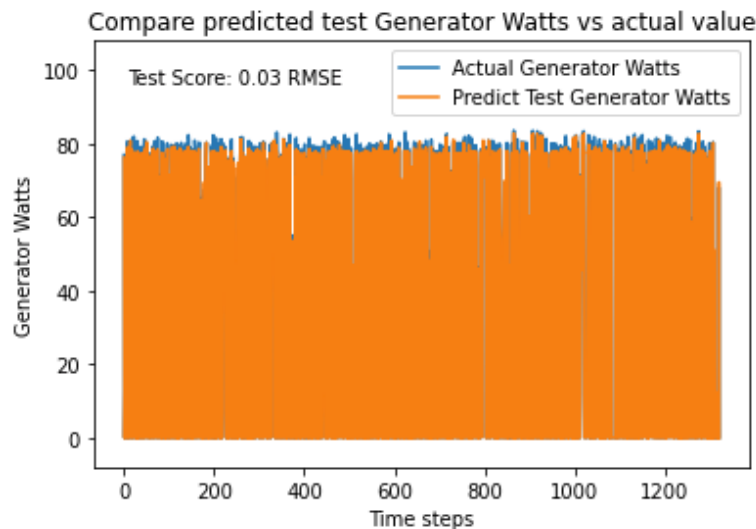
### ***Method four: TOB, Transparent open-box learning network and artificial neural network.***

(This method I learned from a paper after I got this test, it published 2021)

(The detailed explanation about this method can be find in the corresponding code file)

Code explanation: [https://github.com/ShengbinWang/KBO/blob/main/Turbine\\_TOB.ipynb](https://github.com/ShengbinWang/KBO/blob/main/Turbine_TOB.ipynb)

- The predicted value with the actual value within whole timestamps, and the test scores shown in the picture below, it can provide a good prediction from a random input.



- “The transparent open box (TOB) learning network algorithm offers an alternative approach to the lack of transparency provided by most machine-learning algorithms. It provides the exact calculations and relationships among the underlying input variables of the datasets to which it is applied.”

### **Reference:**

- <https://www.sciencedirect.com/science/article/pii/S2405656118300695>
- <https://link.springer.com/content/pdf/10.1007/s42452-020-2249-7.pdf>
- <https://machinelearningmastery.com/sensitivity-analysis-of-dataset-size-vs-model-performance/>