Prediction of CO/NOx emissions from gas turbines using Machine Learning

* By Anshuman Singh (AEE (P), ST - TC&M, Assam Asset)

# Abstract

The flue gas generated from natural gas combustion in gas turbines contains CO and NOx particles, among other pollutants. In this project, a machine learning (ML) model is proposed which can predict the CO/NOx emissions from gas turbines in real-time by taking operational parameters of gas turbines as input variables. This predictive system can be used as a check system to ensure correct functioning of CO and NOX concentrations measurement sensors. The difference between the predictive and measured readings will aid in determining when the CO/NOX concentration measurement sensors are malfunctioning or require re-calibration, thus reducing the chances of excess emissions in the atmosphere. The model’s performance on the test set has shown that this model is able to predict the concentration of CO/NOx emissions in the flue gas with reasonable accuracy and can be implemented in the industry.

# Introduction

Around 69% of the electricity power generated in India during 2021-22 (upto Jan.) was generated from coal as the fuel source. Coal is a highly polluting source of fuel and causes 50-60% more emissions than Natural Gas.

India, with its aim to reduce the total carbon emissions is intending to increase the use of natural gas for power generation by building more gas-fired power plants and infrastructure to supply natural gas.

The flue gas generated from natural gas combustion in gas turbines contains undesirable CO and NOx particles. NOx are serious pollutants which cause photochemical smog, acid rain, ozone layer depletion, and global warming. According to the rules and regulations of the Ministry of Environment, Forest and Climate Change, India, emission of NOx in the flue gas from gas turbines should not exceed 100 mg/Nm3 and is mandated to be monitored continuously using sensors.

As India steadily increases its power generation from natural gas, the pollution resulting from the emissions due to combustion of natural gas in gas turbines will need to be monitored to ensure proper compliance with the regulatory standards and to minimize the damage to our environment.

In this project, we will create a ML model which will be able to predict the CO/NOx emissions in real-time by taking operational parameters of gas turbines as input variables. A ML model is a program that has been trained to recognize patterns in data. The training is done by using a learning algorithm over a training dataset, and once the model is trained, it can be used to make predictions/decisions on new data, assuming that the patterns which were present in the training dataset are also present in the new data.

In this project, we will create a machine learning (ML) model using various learning algorithms from the scikit-learn package.

The data used for training and testing the model is sourced from a publicly available dataset containing data points collected from a CCPP gas turbine plant over 5 years 2011-2015.

This project is built in Jupyter Notebook using scikit-learn, NumPy, pandas and, matplotlib python packages.

# Scope

The ML model proposed in this project is a predictive measurement system to predict the concentration of CO and NOX in the flue gas generated from the gas turbine. This predictive system can be used as a check system to ensure correct functioning of CO and NOX concentrations measurement sensors. The difference between the predictive and measured readings will aid in determining when the CO/NOX concentration measurement sensors are malfunctioning or require re-calibration, thus reducing the chances of excess emissions in the atmosphere.

An automatic alarm system can be created in association with this predictive

measurement system which shall alert the gas turbine operator when the difference between predictive measurements and sensor readings increases beyond a certain threshold.

The predictive system can also act as a backup system for CO/NOX concentration measurement when the CO/NOX concentration measurement sensors are down for maintenance.

# Method

# Data Exploration

The dataset which is used in this project has a total of features 9 features excluding the target variables of CO/NOx (fig. 1).

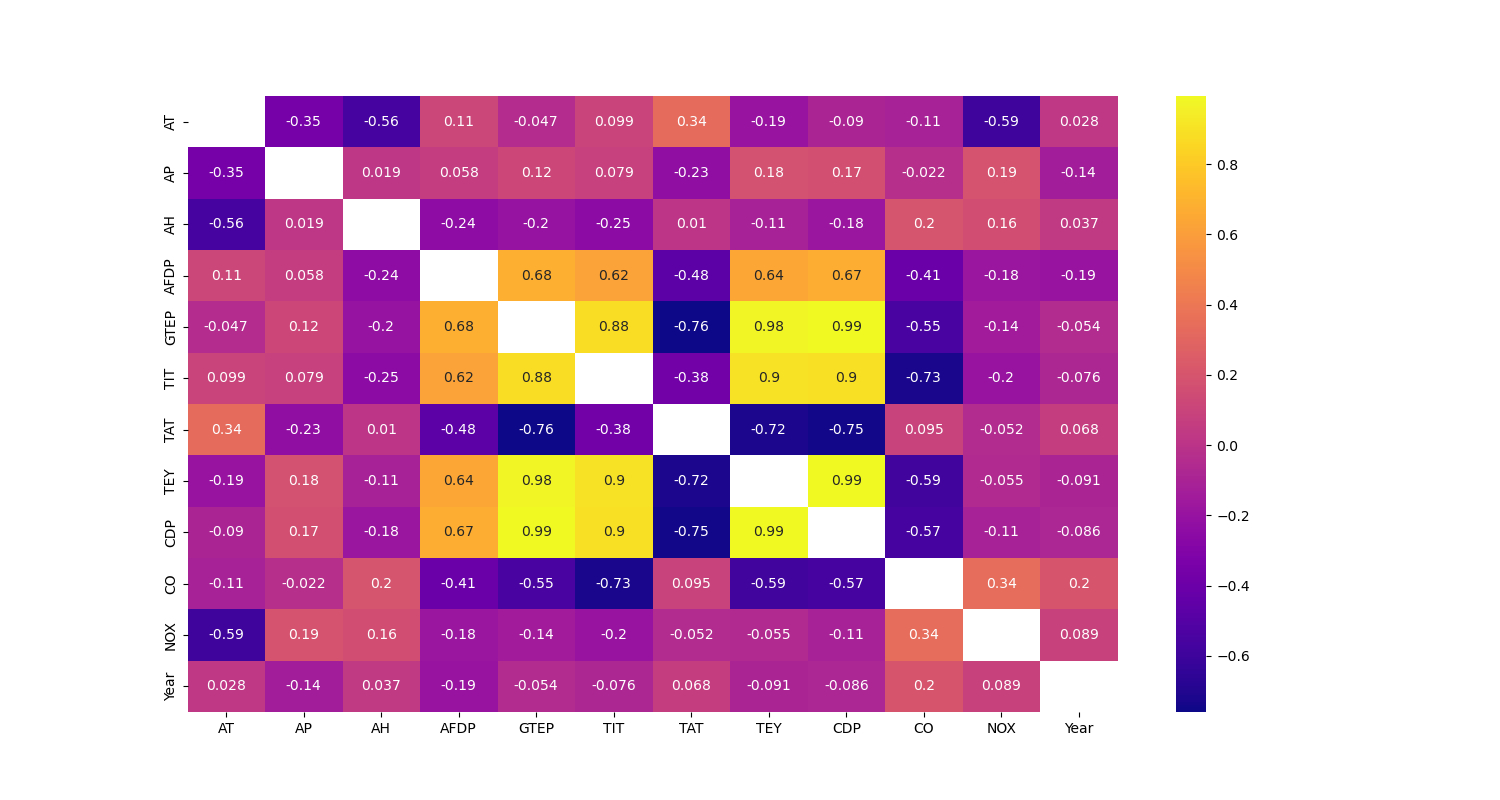
Text

Description automatically generated

Figure 1: Features present in the dataset

Figure 2 contains numerical description of all the features of the dataset. (**count**: number of non-null entries; **mean**: average; **std**: standard deviation; **min**: minimum value; **25% ,50% ,75%**: percentile values at 25%, 50% and 75% respectively; **max**: maximum value). From this table, we can see that the max values for CO and NOX are significantly larger than their average values indicating that some data points have very high CO/NOX emissions. We can see that none of the features have a Null entry as the value of **count** is same for all the features.

The correlation matrix of the features (fig. 3) calculated using Pearson’s Correlation Coefficient is used to determine how the various features are correlated with each other. If the correlation coefficient for two features is positive, it indicates that they are positively correlated. For two positively correlated features, increasing the value of one feature will result in the increase in the value of the other feature, given that all other features are kept constant. The relationship is vice versa if the correlation coefficient is negative.

  
Graphical user interface, application

Description automatically generated with medium confidenceFigure 2: Features Description

Figure 3: Correlation matrix

From figure 3, we can see that GTEP, TET, TIT and CDP are highly correlated with each other. Having highly correlated features is not helpful as they can cause problems when modelling the data using learning algorithms. Thus, only those features should be selected which are important for a particular learning algorithm to identify the patterns present   
in the data. This process of removing features which are detrimental for the performance of the learning algorithm is called feature selection.

# Data Preparation

After exploring the data, outliers from the data were removed before splitting the dataset into training, validation, and testing datasets. Splitting the dataset is necessary because it ensures that the model does not overfit to the training data and generalizes well to new data. To visualize the outliers, we plotted all the features on boxplot diagram (fig. 4).

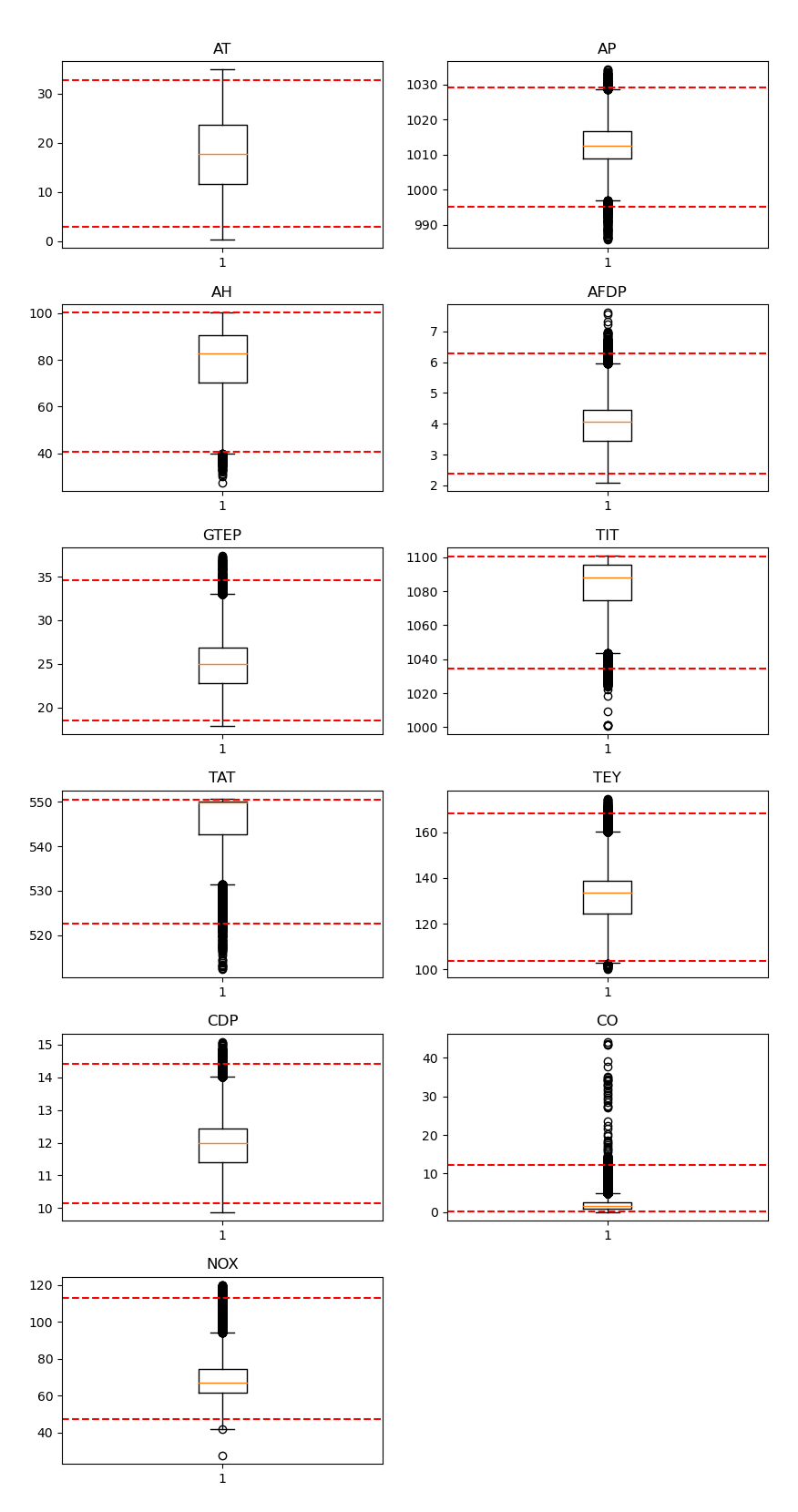


Figure 4: Boxplot distribution of features. The red-coloured dashed lines represent 0.5 and 99.5 percentile values

# 4.3 Modelling and Performance Evaluation

After removing outliers, machine learning algorithms like linear models, Support Vector Machines, Decision Trees, MultiLayer Perceptrons, and K-nearest neighbors were trained on the training dataset and their performance was evaluated on the training dataset itself using cross validation method. The performance measure used in this project is mean absolute error (MAE), which gives the statistical average of the absolute difference of all the predicted values and the actual values. Feature selection and hyperparameter tuning were then applied on all the promising models from the previous step (feature selection is the process of selecting only relevant features to create the machine learning model; hyperparameter tuning refers to the process of finding optimal parameters for the learning algorithm through trial and error).

Subsequently, Voting, Bagging, Boosting (Adaptive and Gradient Boosting) and Stacking techniques were implemented to further boost the performance of the machine learning models.

The performance of all the ML models were then evaluated on the validation set. The performance of the top 4 performing models on the validation set for both CO and NOX predictions are given in table 1 and 2:

Table 1: Mean Absolute Error scores for CO prediction on the validation set for top 4 models

|  |  |
| --- | --- |
| MAE (Mean Absolute Error) for CO prediction | |
| Gradient Boosting | 0.754 |
| Adaptive Boosting on Random Forests | 0.775 |
| Voting Regressor on Random Forest + Multilayer Perceptrons + K-Nearest Neighbors | 0.723 |
| Stacking Model (Layer 1: Random Forests + Multilayer Perceptrons + K-Nearest Neighbors; Layer 2: K-Nearest Neighbors) | 0.746 |

Table 2: Mean Absolute Error scores for NOX prediction on the validation set for top 4 models

|  |  |
| --- | --- |
| MAE (Mean Absolute Error) for NOX prediction | |
| Voting Regressor on Random Forest + Multilayer Perceptrons + K-Nearest Neighbors | 4.985 |
| Random Forests | 5.100 |
| Multilayer Perceptrons | 4.977 |
| Stacking Model (Layer 1: Random Forests + Multilayer Perceptrons + K-Nearest Neighbors; Layer 2: K-Nearest Neighbors) | 5.147 |

# Results and conclusion

Adaptive Boosting on Random Forests performed best on the test dataset for prediction of CO and was able to achieve a mean absolute error of 0.84 (fig. 5)

Voting Regressor on Random Forest + Multilayer Perceptrons + K-Nearest Neighbors performed best for prediction of NOX on the test dataset and was able to achieve a mean absolute error of 11.32 (fig. 6)

The model’s performance on the test set has shown that this model is able to predict the concentration of CO/NOx emissions in the flue gas with reasonable accuracy and is ready for implementation in the industry. The model is also computationally less expensive as compared to simulation models due to low CPU requirements. Moreover, once implemented, the model’s performance will get better with time as it will get progressively more data to learn from, thus further increasing the accuracy of the model.

Chart, scatter chart

Description automatically generated

Figure 5: Adaptive Boosting on Random Forests for prediction of CO (performance on training data (left), validation data (middle) and test data (right))

Chart, scatter chart

Description automatically generated

Figure 6: Voting Regressor on Random Forest + Multilayer Perceptrons + K-Nearest Neighbors for prediction of NOX (performance on training data (left), validation data (middle) and test data (right))

# References

1. Link to the dataset used in this project: <https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set>
2. J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
3. [Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.