Data Analysis in Process Industries

A work on Gas sensor array's output under dynamic gas mixtures

Design Oriented Project
Project Report

Submitted by:

Reventh Sharma	2017A1PS0832P
Aashish Aggarwal	2016B3A10578P
Somya Agrawal	2016B1A10741P

Under the able guidance of:

Prof. Suresh Gupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Nov 2019

ABSTRACT

In the following paper, the methodology to solve the problem of predicting concentration in a gaseous mixture from sensor array output has been discussed. For this, a time variate analysis algorithm to detect anomalous values have been derived.

Various machine learning algorithms available have been analyzed to predict the algorithm best suited for prediction. This methodology can be used in process industries where the streams are controlled to have specific concentrations. Hence the algorithms only predict concentration after the sensor is exposed to stream so that they get saturated with target gas. This method is proved to be more accurate in determining gas concentrations in the system when sensor have approached an equilibrium with feed gas.

TABLE OF CONTENTS

INTRODUCTION

In the real-world, processes in industries involve many intricacies and complex work. With the development of technology, attempts have been made to control dynamic variables such as flow rate, composition, temperature and pressure of streams and compounds in process equipment. Sensors enable us to observe all such details while the procedure is getting executed and accordingly changes can be made so that the desired product is obtained. MOX based sensor is a very efficient class of sensors to detect the concentration of gaseous components in a stream or a vessel. To add more parameters for efficient detection of the concentration of different species array of MOX based sensors is generally used.

MOX sensors have 1 to 3 gas-sensitive metal oxide layers. This makes them sensitive to gas and they respond to exposure to gas by undergoing a change in the conductivity of their semiconductor layer/s. The gas-sensitive layer of a specific MOX sensor reacts to reducing gases with increasing of the layer resistance and to oxidizing gases with decreasing of the layer resistance. Changes in the resistance of the sensors can be measured by measuring the change of capacitance, work function, mass, optical characteristics or reaction energy. The type of the MOX sensor and the types of gases decide the concentration range of gases in which the sensor(layer/s) will react. A MOX based gas sensor array consists of sensors of different types. These different types of sensors respond to different concentrations of gases and thus, change in concentration of gases can be done more efficiently because of increased parameters. The sensitivity of these sensors can be improved by using a specific catalyst in the semiconductor layers. The response time of the gas sensor array depends on the volume of the measurement chamber and the flow rate.

LITERATURE REVIEW

As the sensor array generally provides an output based on a change in their conduction, the voltage across them or current passed through them hence the thermodynamic variables need to be predicted from the sensor output. This process can be easily done where a single variable needs to be predicted, but when more than one variables are to be predicted we need a complex set of algorithms to do this task.

Predicting the composition of various components in a stream or vessel is a primary task to control a process in a plant. Hence, in the given study we tried

predicting the methodology which is best suited for the aforementioned task. For this, we have used the dataset available on UCI Repository as the title "Gas sensor array under dynamic gas mixture" dataset[1]. Previous work done on the given dataset includes predicting concentration under a dynamic condition where concentrations of species can change randomly using the Reserve Computing algorithm[2]. However, in a process plant, the stream concentrations are generally fixed in a controlled manner and are required to be known with very high accuracy. Hence a method is devised here so that the same can be predicted more accurately. We have tried to devise a method to predict anomalous value during the runtime of the sensor array. For the prediction of gas concentration, instead of devising a new algorithm we have tried to focus on existing literature of algorithm to predict the most efficient way to get accurate predictions. Work has been done to predict ethylene and CO and ethylene and methane concentrations separately. The algorithms are applied to Python and data visualization is done through MATLAB. The following are the algorithms that have been used in our work.

Linear Regression

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables *(x)* and the single output variable (*y*). More specifically, *y* can be calculated from a linear combination of the input variables (*x*). It is a supervised ML algorithm. When training the model – it fits the best line to predict the value of *y* for a given value of *x*. The training results in the best values for different θ with the help of the best fit line.

$$Y = \theta_0 + \sum_{i=1}^{n} \theta_i X_i$$

Cost Function (J):

By achieving the best-fit regression line, the model aims to predict *y* value such that the error difference between the predicted value and the true value is minimum. So, it is very important to update the θ values, to reach the best value that minimizes the error between *predicted y* value *(pred)* and *true y* value *(y)*.

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2 \qquad J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

Cost function (*J*) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (*pred*) and true y value (*y*).

Gradient Descent:
To update θ values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ values and then iteratively updating the values, reaching minimum cost.

Polynomial Regression

Polynomial Regression is a form of linear regression in which the relationship between the independent variable *x* and dependent variable *y* is modelled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of *x* and the corresponding conditional mean of *y*, denoted *E(y |x)*.

$$Y = \theta_0 + \sum_{i=1}^{n} \theta_i X_i{}^i$$

Since regression function is linear in terms of unknown variables, hence these models are linear from the point of estimation. Polynomial regression is applied where the model to be fitted may be showing curvilinear trends. Though, polynomial regression provides a better fit to data, using a much higher polynomial degree may result in overfitting of the data.

Neural Network

An artificial neural network learning algorithm is a computational learning system that uses a network of functions to understand and translate a data input of one form into the desired output, usually in another form.

The neural net learning algorithm learns from processing many labelled examples (i.e. data with "answers") that are supplied during training and using this answer key to learn what characteristics of the input are needed to construct the correct output. Once a sufficient number of examples have been processed, the neural network can begin to process new, unseen inputs and successfully return accurate results. The more examples and variety of inputs the program sees, the more accurate the results typically become because the program learns with experience.

Neural networks can be applied to a broad range of problems and can assess many different types of input, including images, videos, files, databases, and more. They also do not require explicit programming to interpret the content of those inputs.

Because of the generalized approach to problem-solving that neural networks offer, there is virtually no limit to the areas that this technique can be applied.
A typical structure of single layer neural network is given in fig.1

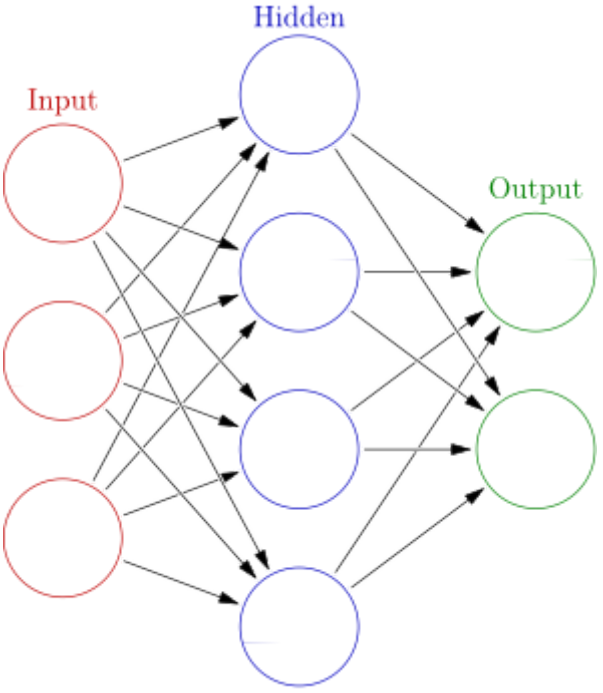


Fig.1

Decision Tree

Decision trees are one of the common supervised learning algorithms which are used for building classification and regression models in the form of a tree. It breaks the dataset into smaller subsets which increases the depth of the tree. The depth of the tree decides the complexity of the tree and the accuracy of the model. The final tree developed using this algorithm has two types of nodes - decision nodes and leaf nodes. A decision node has two or more than two branches whereas a leaf node represents a classification or decision. The topmost decision node in a tree corresponds to the best predictor and is called the root node.
Decision trees can handle both categorical and numerical data. It can also handle multi-output problems. They are drawn upside down with the root at the top of the tree. Various steps are involved in making a decision tree. First is splitting

which involves partitioning of the data set into subsets. All the features are considered and splits are made. Cost functions are used to select the best split. The split with the lowest cost is considered. This algorithm is recursive in nature. It is also important to know when to stop splitting as otherwise this algorithm will lead to very complex trees in problems having large number of features which can lead to overfitting of the data To stop splitting , sometimes a minimum number is set for training inputs to be used on each leaf or a maximum depth of the tree (model) is set which refers to the length of the longest path from a root to a leaf. Another step involved in developing a decision tree is pruning. Pruning is a method of removal of branches that are made from less important features. It helps in reducing the complexity of the tree and thus removes the chances of overfitting. One simpler pruning technique is reduced error pruning in which the nodes at the branched are removed if that does not affect the accuracy of the model. A more sophisticated way of pruning is cost complexity pruning where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree. This is also known as weakest link pruning. Decision trees implicitly perform variable screening or feature selection. Nonlinear relationships between parameters do not affect tree performance.

Random Forest

Random forest is a supervised learning algorithm that consists of a large number of decision trees. A decision tree is the basic building block of a random forest Each decision tree in the forest considers a random subset of features when forming questions and has access to only a random set of the training data points. Features for every tree are selected randomly so as to avoid any correlation between the decision trees. This algorithm can also be used for both classification and regression problems. This algorithm involves two stages or steps. First is, creation of the forest and the second is, making a prediction from the random forest classifier created in the first stage. The algorithm works in this way: out of the total m features, k features are selected such that $k << m$. Among the k features, the node d is calculated using the best split point. The node is split into daughter nodes using the best split. This is done until we get l number of nodes. All these steps are repeated n number of times t get n number of decision trees. This gives us a random forest classifier. Now, to make predictions test features are taken and each decision tree is used to predict the outcome. The predicted outcome is stored and the outcome which is most predicted by the decision trees in the random forest is considered to be the final prediction from the random tree algorithm.

METHODOLOGY

The 16 sensor values given in the datasets are for the corresponding concentrations of the two different species present in the mixture at different blocks of time of the same interval. The concentrations were abruptly changed and the values of the sensor were tabulated. We use this tabulated data to train and test the aforementioned algorithms after preprocessing the data to remove discrepancies and get better results.

The dataset contained more than ~42Lakh observations and the samples were collected with a sampling frequency of 0.01 seconds for a total period of 12 hours. This data was divided into blocks of time of 100 seconds each in a way such that the concentrations of one or both the components were changed abruptly to a random value.

The change in sensor value when exposed to particular concentrations of gas mixture of ethylene and CO and ethylene and methane are given in fig.2 and fig.3 respectively. The gas concentrations correspond to that between t=800s to t=900s in the dataset.

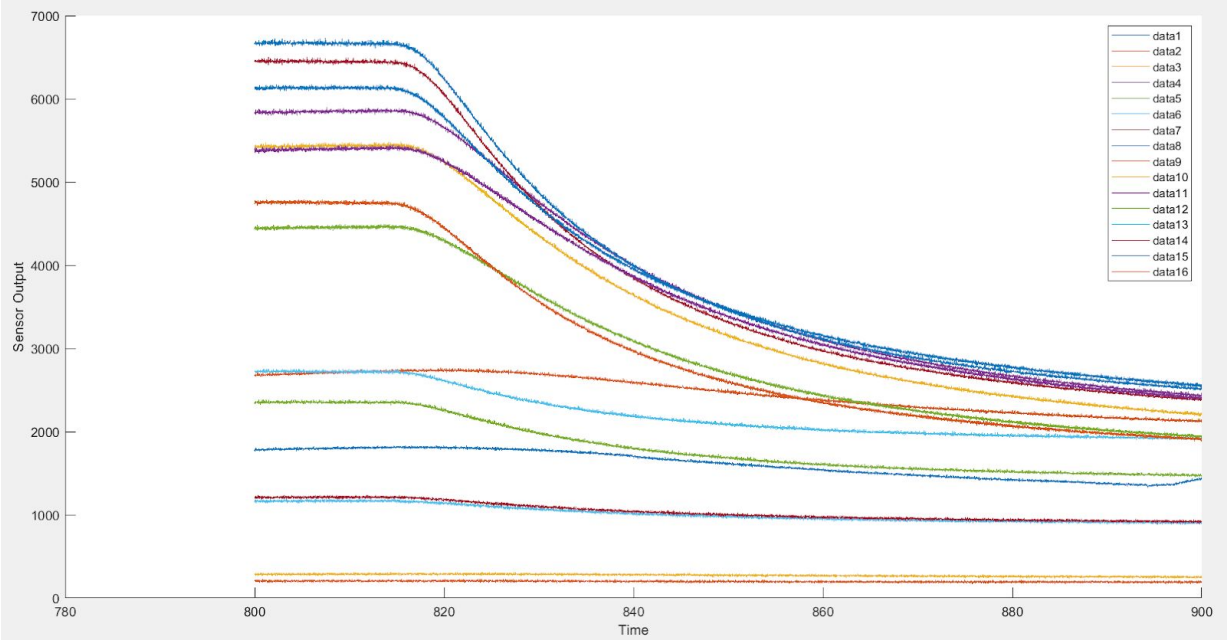


Fig.2

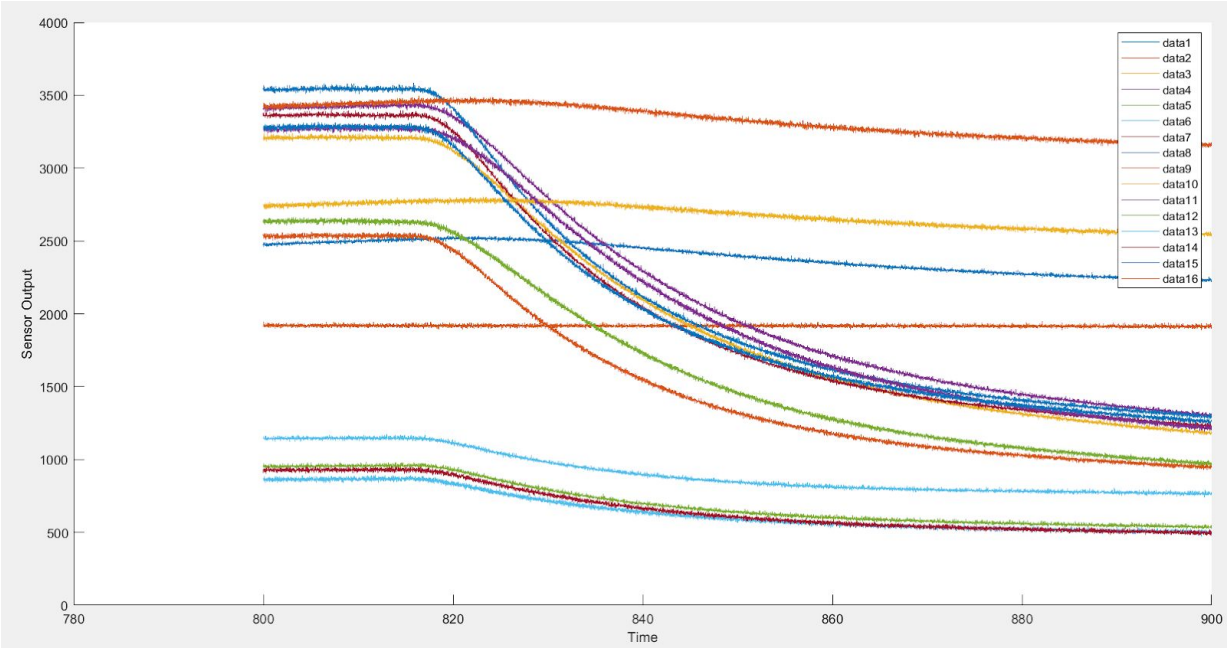


Fig.3

From the figures it can be observed that the sensor values are seen to approach steady values after approximately 80 seconds. Hence, it can be concluded that the equilibrium time for the sensors is 80 seconds. The concentrations are not changed at exactly 100 seconds and there exist intervals in which concentrations are changed before 100 seconds. However, there doesn't seem to exist any interval in which the concentrations are changed before 99 sec. Hence our time range for data analysis consist of time interval between 80 to 99 seconds for each block.

Anomaly Detection

The sensors can behave anomalously during their entire run. Hence, it is important for us to remove these anomalous values with more standardized values which are expected to occur according to running trend for accurate prediction.

After equilibrium time the sensor values for particular concentrations are normally distributed w.r.t their occurrence frequency in intervals of 2.5sec after they reach equilibrium wrt supplied concentration.. Hence the next sensor value must not deviate more than 4σ (probability of occurrence in interval is 99.98%) from previous value. The standard deviation is calculated for intervals of 2.5 seconds wrt to latest sensor value. If the new value deviates more than 4σ it is considered anomalous and hence is replaced by 'previous value + 4σ'. Due to the dynamicity of this algorithm it can be used in continuous time series prediction after equilibrium time of sensor is reached.

However for sensor-2 a particular anomalous peaks occur after 5000seconds of sensor run. These peaks don't get standardized by the algorithm as very high values are approached suddenly. Hence, the peaks are trimmed to the mean value of their of preciding 2.5 seconds of sensor data.

Linear Regressor

In this study, out of the algorithms used, linear regression has the limitation of resulting in a single output. Hence, to work around this issue, when training the linear regressor, one of the two concentrations was randomly assumed to be zero. With this modification, the linear regression was used to predict one of the two concentrations for different blocks of time.

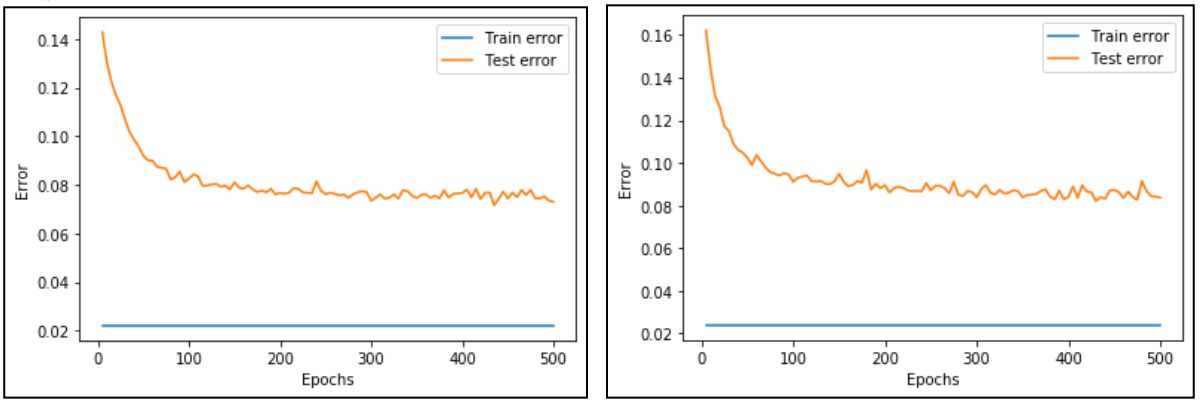
Polynomial Regression

Polynomial regression was used, assuming, to predict the data better and increase the accuracy as compared to linear regression such that a higher degree polynomial would be better able to fit the data. For our study, the highest degree taken was of order 3.

Neural Networks

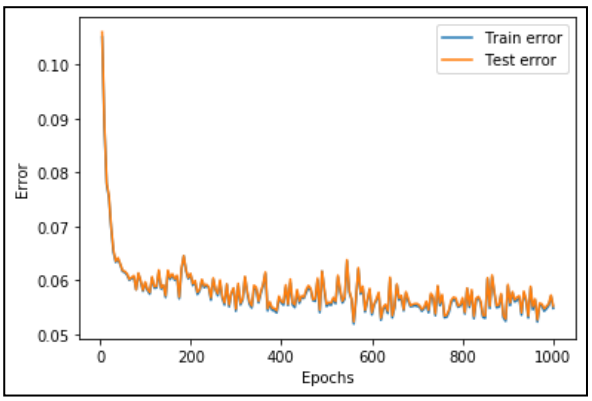
Neural Network, being an advanced complex algorithm is able to predict multiple outputs with better accuracy than the basic algorithms. In this study, a deep layer network architecture of 1input (16 nodes), 3 hidden (50*100*10 nodes) and 1 output (2 nodes) layers have been used with the number of epochs ranging from 1 to 500. The test and train error for CO and Ethylene prediction are plotted wrt number of epochs in fig.4 and Ethylene and Methane are plotted in fig.5.

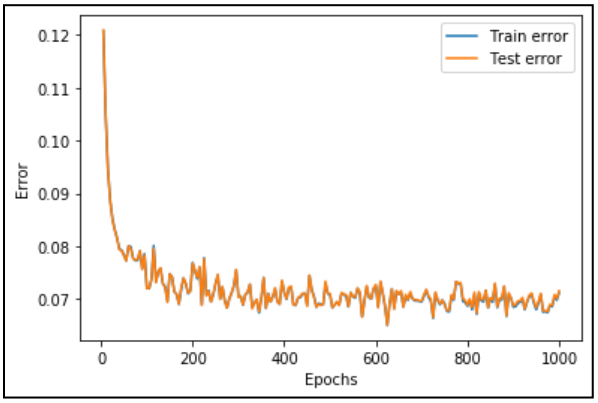
Hence, the data is trained 430 times for ethylene-CO dataset and 610 times for ethylene-methane dataset.



CO Ethylene

Fig.4





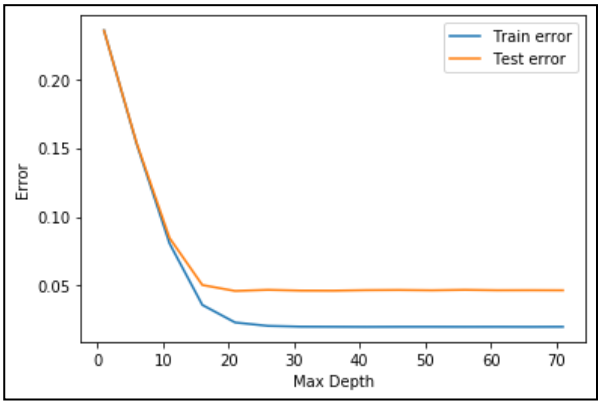
Methane Ethylene

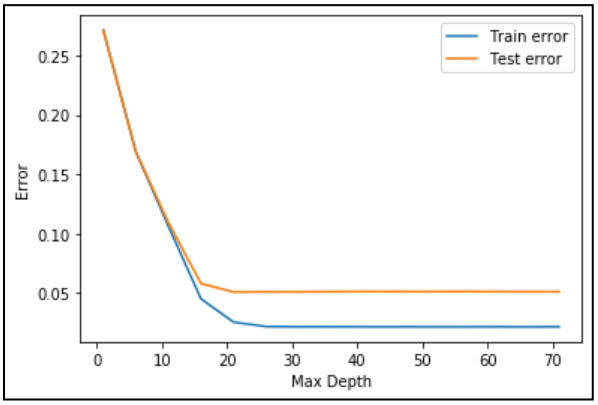
Fig.5.

Decision Trees

The decision tree is the most fastest algorithm to get trained on a given machine. The training error for decision trees get reduced as the depth of the tree is increased. However, increasing depth often leads to overfitting. Hence, to analyse the depth after which overprediction can occur we plot the test and train error for both the datasets w.r.t depth of tree in fig.6 and fig.7 respectively. For each node in the tree, minimum 100 samples are to be required in the node for it to get split and each node after splitting should consist of at least 20 samples.

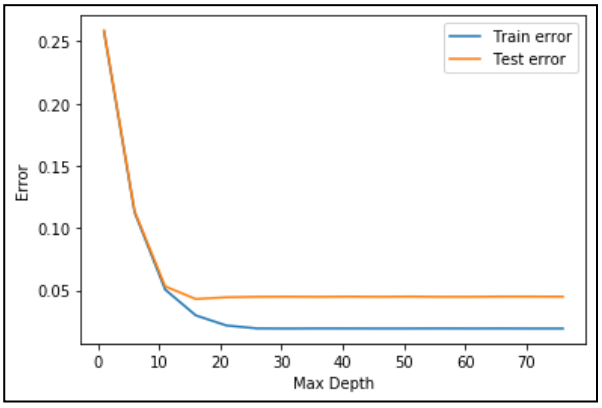
Hence the max depth for Ethylene-CO dataset is taken as 36 and that for ethylene-methane dataset is taken as 56.

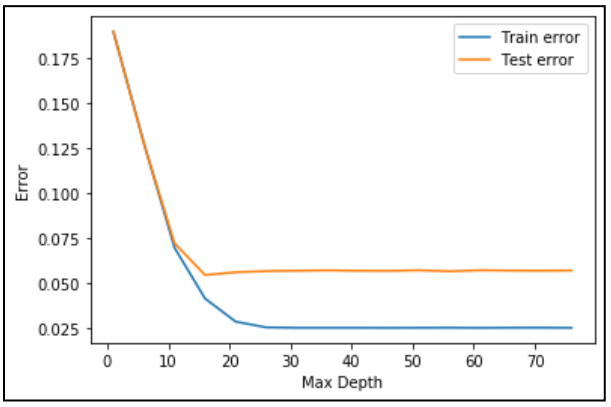




CO Ethylene

Fig.6.





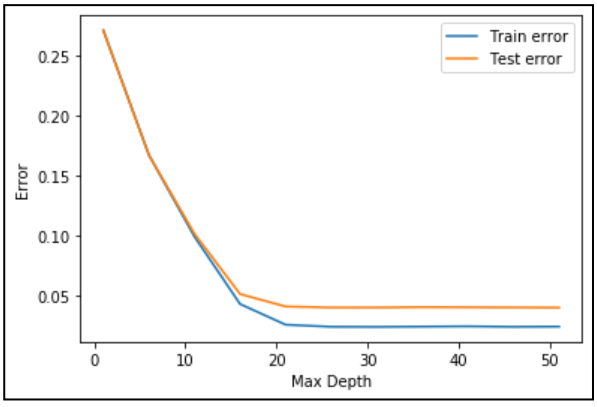
Methane Ethylene

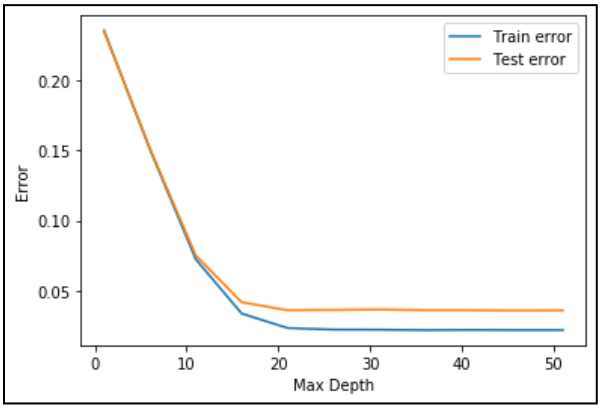
Fig.7.

Random Forests

The random forest make prediction by taking the mean of values predicted from 100 trees. Hence accuracy is increased in random forest. It suffers from the same issue of overprediction hence train and test error are plotted w.r.t tree depth in fig.8 and fig.9 for the two datasets. For each node in the tree, minimum 100 samples are to be required in the node for it to get split and each node after splitting should consist of at least 20 samples. The random forest takes much larger time than decision tree for training as 100 decision trees are needed to be trained simultaneously in it.

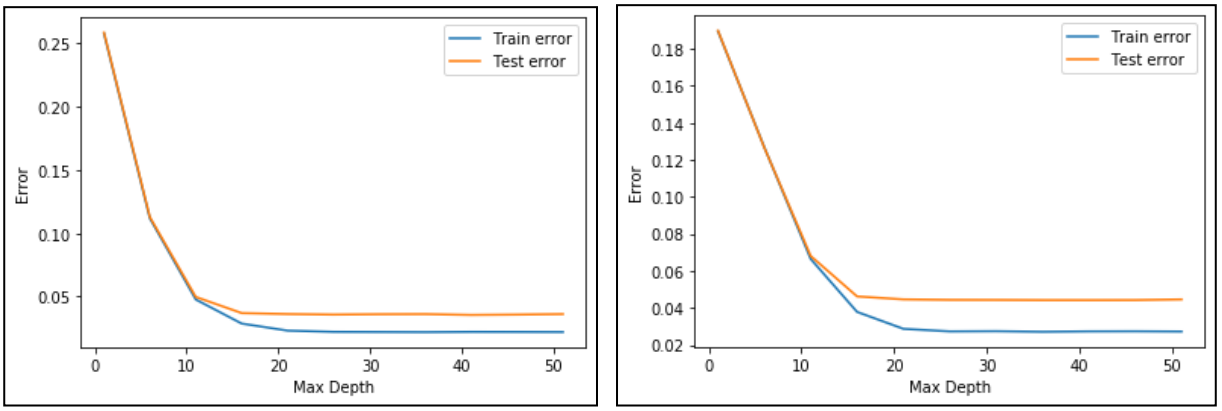
Hence, the max depth taken for Ethylene-CO dataset is 36 and that for ethylene-methane dataset is taken as 40.





CO Ethylene

Fig.8.



Ethylene Methane

Fig.9.

ANALYSIS AND RESULTS

The RMS errors when the algorithms were applied on the test dataset for ethylene_CO prediction and ethylene_methane predictions are given in table-1.

Algorithm	$RMSE_{CO}$	$RMSE_{Ethylene}$	$RMSE_{Methane}$	$RMSE_{Ethylene}$
Linear	94.535	4.036	43.919	3.352
Polynomial	3.523	3.523	35.906	2.781
Decision Tree	25.581	1.075	12.471	1.075
Random Forest	19.276	0.795	10.635	0.892
Neural Network	38.565	1.644	16.347	1.334

Table-1

The maximum CO concentration in the dataset was 533.33ppm and that of ethylene is 20ppm in the ethylene-co dataset. The maximum methane concentration in the dataset was 296.67ppm and that of ethylene is 20ppm in the ethylene-methane dataset.

CONCLUSION

It has been seen that the polynomial regression was most resource intensive as it expanded the data size to three times of its initial value (for degree 3). This resource intensiveness had minimum cost benefit as only one concentration could be predicted by the algorithm. Though very much efficient, this efficiency was only the result of predicting only one concentration from 16 input sensor values.
The linear regression was least efficient as the sensor output doesn't vary with concentration linearly.

Amongst the algorithms predicting both the concentrations in gases Neural Network took the highest amount of time to train while the decision tree took the smallest. Decision tree also had the least error after random forest. Hence, it is recommended to use decision tree algorithm for predicting concentrations using MOX-sensors in process industries.

REFERENCES

1. Fonollosa, J., Sheik, S., Huerta, R., & Marco, S. (2015). Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators B: Chemical*, *215*, 618–629. DOI: 10.1016/j.snb.2015.03.028

2. Liu, X., Cheng, S., Liu, H., Hu, S., Zhang, D., & Ning, H. (2012). A Survey on Gas Sensing Technology. Sensors, 12(7), 9635–9665. DOI:10.3390/s120709635

3. Dey, A. (2016). Machine learning algorithms: a review. International Journal of Computer Science and Information Technologies, 7(3), 1174-1179.

4. Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168). ACM.

5. D. Opitz, R. Maclin, "Popular Ensemble Methods: An Empirical Study", Journal of Artificial Intelligence Research, 11, Pages 169-198, 1999