American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-10, Issue-12, pp-19-33
www.ajer.org
Research Paper Open Access

Comparative Evaluation of Machine Learning Regression Algorithms for PM2.5 Monitoring

Fidelis C. Obodoeze[1], Chris A. Nwabueze[2], Silas A. Akaneme[3]
[1](Doctoral Research Candidate, Department of Electrical and Electronic Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli, Nigeria,
Corresponding Author)
[2](Professor, Department of Electrical and Electronic Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli, Nigeria)
(Assistant Professor, Department of Electrical and Electronic Engineering, Chukwuemeka Odumegwu Ojukwu University, Uli, Nigeria)

ABSTRACT : *Air pollution is a huge challenge to the residents of highly populated cities and municipal managers over the years because of the serious threats it poses to human health and environment. Machine learning and deep learning are branches of Artificial Intelligence (AI) that can be used to train past historical dataset to identify patterns in an occurrence in the dataset which can be used to predict or forecast future occurrences of air pollution in that particular location. In this paper, particulate matter (PM2.5) historical datasets of a smart city with accompanying meteorological datasets from 2008 to 2013 from the city of Beijing China was used to carry our experimental evaluations of the performances of nine different machine learning algorithms including five (5) traditional machine learning algorithms such as Multi Linear Regression (MLR), Multi Layer Perceptron Neural Network (MLP-ANN), Support Vector Regressor (SVR), Decision Trees , Lasso and four ensemble algorithms such as Extra Trees, Extreme Gradient Boosting (XGBoost), Random Forest, Boosted Decision Tree (Decision Tree boosted with AdaBoost) in terms accuracy of prediction, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), in terms of predicting the future PM2.5 concentrations in the smart city. The experimental results showed that weather or meteorological parameters such as temperature, air pressure, relative humidity, windspeed and rainfall or precipitation have a direct correlation or effect on the successful emission and prediction of PM2.5 air pollutant concentrations in a particular city or location. Finally, the experimental result showed that ensemble machine learning algorithms such as XGBoost, Extra Trees, AdaBoost and Random Forest outperformed other traditional machine learning algorithms in terms of prediction accuracy and reduced prediction errors. XGBoost model scored the highest R2 score of 0.853 followed by Extra Trees (R2=0.852), Boosted Decision Tree (Decision Trees +AdaBoost) came 3rd with R2=0.850), then followed by Random Forest (R2=0.847) while traditional machine learning algorithm such MLP Neural Network (MLP-ANN) came 5th with R2=0.838, followed by Lasso algorithm and Linear Regression (MLR) both having R2=0.812, followed by SVR with R2=0.767 and Decision Trees came last with R2=0.683 in accuracy of prediction.*

KEYWORDS: *Particulate matters, Air pollution, Air Quality Index, Regression Analysis, time-series data, Meteorological dataset*

Date of Submission: 02-12-2021 Date of acceptance: 15-12-2021

I. INTRODUCTION

Air pollution is a major challenge in municipal and city administration because of the adverse effects it poses to the health and comfort of the residents of the city. Health effects of air pollution and pollutants are enormous and include increase in respiratory and cardiovascular diseases such as asthma, pneumonia, bronchitis, COPD, laryngitis, heart diseases and even caner. Air pollution monitoring stations or sub-stations can be installed or deployed in various locations within a city to monitor in real time the concentrations of this unwanted air pollutants. Unfortunately, not many cities can afford costly air quality monitoring stations or sub-stations because high costs of full air quality monitoring stations. Air quality prediction systems easily come to

the rescue because they can give accurate and reliable air quality forecast in advance to city managers or members of the public as would be obtainable with real-time air quality stations and monitors. Air quality prediction systems can employ machine learning and deep learning algorithms and models to predict the possible outcomes of air quality of a city in advance using historical weather or meteorological data and past air pollutant concentrations of the particular air pollutant to measure [1].

PM2.5 pollutant concentrations and other air pollutants can be modeled using either chemical models or data-driven models but chemical models (i.e. classical deterministic models, sometimes referred to as chemistry-transport models) are based on the chemical laws to model all the relevant chemical processes that contribute to PM2.5 formation. Such models may describe up to hundreds of species such as troposphere, photochemistry and aerosols [2].

Data-driven models also known as stochastic approaches use historical data to make future predictions. They are based specifically on statistical approaches. Chemical transformation models for air pollution are very complex, and the current detailed list of emissions is very difficult to obtain [2]. However, there are lots of limitations in the prediction of PM2.5 concentrations using chemical models. Such limitations include:- (1.) that model inputs such as the emission inventories are not very accurate (for spatial and time distribution, for chemical speciation, i.e. analytical method in identifying and/or measuring the quantities of one or more individual chemical species in a sample), the meteorological fields are also uncertain, etc. (2). There may be high mathematical and computational burden as a result of too many numbers of parameters [2].

In this paper, data-driven models will be used to model PM2.5 pollutant concentrations. Data-driven approaches such as supervised machine learning and deep learning algorithms have been used in literature and in smart city air pollution or air quality prediction and they include the following – Multi Layer Perceptron Feed Forward Artificial Neural Network (MLP- ANN), Radial Basis Function (RBF), Support Vector Machine Regressor (SVR), Linear Regression, Lasso, Convoluted Neural Network (CNN) e.g. Long Short Last Term (LSTM), Naves Bayes and Decision Trees). Ensemble algorithms such as Random Forest (RF), Extreme Gradient Boosting algorithm (XGBoost), AdaBoost, Extra Trees and hybrid methods, etc.

Python programming has become the most popular tools for air pollution prediction and forecasting in recent times as a result of its wide support for Artificial Intelligence (AI), machine and deep learning algorithms that are inbuilt into its libraries such as Tensorflow, Keras, Scikitlearn, etc. Python programming language is also an open source programming paradigm that allows easy access and modification of the source codes by the programmers. Apart from this, it has a huge community of programmers and researchers worldwide. Python programming language version 3 Machine Learning modules was used for machine learning algorithms with Anaconda and Jupyter Notebook Integrated Development Environment and platforms as experimental testbed extensively to develop these models.

II. REVIEW OF RELATED LITERATURE

Emina Džaferovic and Kanita K. Hadžiabdic [3] presented a research paper on Air quality prediction of the city of Bjelave, Sarajevo using different machine learning algorithms – Support Vector Regression (SVR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Multi Layer Regression (MLR) and Multi Layer Perceptron (MLP) ANN using three years of historical dataset (2016-2018) comprising of five air pollutant concentrations of PM10, NO2, SO2, O3 and CO together with eight (8) meteorological parameters – minimum temperature, maximum temperature, average temperature, wind speed, wind direction, humidity, pressure and precipitation. Experimental results showed that Random Forest (RF) regression algorithm outperformed other machine learning algorithms in terms of the highest R2 and lowest RMSE, closely followed by XGBoost algorithm.

Aceves-Fernández et al [4] presented a research paper on evaluation of key parameters using Deep Neural Convolutional Neural Network (CNN) for airborne pollution prediction to forecast or predict PM10 concentrations in the air based on atmospheric variables. The authors used a method known as Bagging Ensemble Method (BEM) to improve the accuracy of the prediction of the model while CNN with (1D and 2D) were explored to develop the model. The meteorological variables selected in the experiment included temperature (TMP), Wind direction (WDR), Wind Speed (WSP), Relative Humidity (RH), Solar Ultraviolet radiation type A (UVA) and Solar Ultraviolet radiation type B (UVB) alongside an hourly pollution dataset from a public database of the Mexican Ministry of Environmental Agency from 2010 to 2018. Experimental results showed that the Bagging Ensemble Method (BEM) increased the accuracy of the CNN prediction model by 20%.

Bingyne Pan [5] applied Extreme Gradient Boosting algorithm (XGBoost) to predict PM2.5 concentrations on hourly basis in the city of Tianjin, China. The author made use of historical dataset of PM2.5 concentrations (December 1, 2016 – December 30, 2016) of about Nineteen air pollution monitoring stations. The set of input variables or parameters in the modeling include hourly concentrations of PM2.5, SO2, NO2, CO and Ozone (O3). The dataset was about 6845 samples of historical dataset altogether. XGBoost algorithm

was compared to other machine learning (ML) algorithms in the experiments. The experimental results showed that XGBoost algorithm outperformed other ML algorithms such as Random Forest (RF), Multiple Linear Regression (MLR), Decision Trees (DT), and Support Vector Regression (SVR) in terms of prediction accuracy(i.e. highest R2 value) and lowest error value (i.e. MAE and RMSE).

Kaya & GündüzÖğüdücü[6] worked on air pollution index modeling and prediction using a hybrid method of Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) to predict/forecast the concentrations of PM2.5 in Istanbul Turkey between 2014 and 2018 in 4, 12 and 24 hourly basis in advance. Inputs to the model include other air pollutant parameters such as CO, NO, NO2, NOX, O3 and SO2 collected from the closest air pollution monitoring stations. Equally meteorological variables such as maximum temperature, minimum temperature, wind speed, wind direction, maximum wind speed, maximum wind direction, and humidity were also added as input dataset. The meteorological parameters were collected on hourly basis from the online real-time service belonging to the Turkish State Meteorological Service (TSMS). The experimental results showed high degree of prediction accuracy using the two combined or hybrid methods. Qadeer et al. (2020) worked on the prediction of the particulate matter PM2.5 concentration for two cities in South Korea using ANN Long Short Term Memory (LSTM) network. They used 24-hour data of 16 input predictors of measured air pollutant concentrations as well as meteorological data to predict the next 1 hour of PM2.5 concentration. The four metrics were used such as MAE, mean squared error (MSE), etc. to validate or evaluate the obtained results. The prediction performance experiment was carried out using five other models. The result of the experiment confirmed that the LSTM outperformed other models in terms of accuracy in predicting the PM2.5 concentration.

Joharestani et al [7] presented a research work on the prediction of PM2.5 concentrations using three machine learning algorithms namely extreme gradient boosting (XGBoost), Random Forest and Deep Learning algorithm (LSTM) to predict hourly PM2.5 in the city of Tehran, Iran. Historical dataset of air pollutants such as PM2.5, PM10, CO, Ozone(O3), NO2, and SO2 (from January1, 2015 to December 31, 2018) as well as meteorological dataset consisting of air temperature, maximum and minimum air temperature , relative humidity (RH), daily rainfall, visibility, wind speed, sustained wind speed , air pressure, and dew point were used as input predictors. Also, satellite data known as Aerosol Optical Depth (AOD) was used as an input predictor or parameter. Other input parameters used include Day of the year, Day of the week, Season, longitude and latitude. From the experiments conducted there was evidence that XGBoost algorithm performed best compared to Random Forest and Deep learning algorithms such as CNN and LSTM in terms of speed of prediction (19s), highest prediction accuracy (R2=0.81, R=0.9), lowest prediction errors (RMSE=13.58 µg/m3, MAE=9.92µg/m3). It was also observed that satellite data did not have any significant effect on the prediction accuracy when introduced into the modeling.

III. MATERIALS AND METHODS

The reviewed works employed different machine and deep learning algorithms. The research approach is to conduct a regression prediction experiment to evaluate the best machine and deep learning algorithm in terms of performance accuracy before employing a particular algorithm for air pollution prediction.

3.1 Regression And Estimation Models

Regression prediction machine learning algorithms employed are Multi Linear Regression (MLR), Support Vector Machine Regression (SVR), Multi Layer Perceptron (MLP) Feed Forward Artificial Neural Network (MLP-ANN), Decision Trees (DT), Extra Trees, Boosted Decision Trees (i.e. Decision Tree boosted with AdaBoost), Random Forest (RF), Extreme Gradient Boosting algorithm (XGBoost), and Lasso regression algorithm.

3.1.1 Regression Techniques

3.1.2 Linear Regression

Linear Regression is a very popular and commonly used regression model developed in the field of statistics for carrying out predictive modeling primarily concerned with minimizing the error of a model or making the most accurate predictions possible. It is used in machine learning to prepare or train the regression model equation from dataset. Assuming there is a single input (X), the method is referred to as Simple Linear Regression (SLR). When there are more than one or multiple input or predictor variables, it is referred to as Multiple Linear Regression (MLR).

The formula for MLR can be represented as Hayes [8].

$$Y=\beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 \ldots\ldots+\beta_{p-1}X_{p-1} + \varepsilon \quad (1)$$

Where Y is the output or dependent variable; X_1, X_2, X_3,.. X_{p-1}, are the input or predictor or independent variables; β_0, β_1 β_2 β_3 β_{p-1} are regression coefficients for each input variables and ε is the model residual or random errors.

3.1.3 Multi Layer Perceptron Neural Network (MLP NN)

Multi Layer Perceptron (MLP) is a supplement of feed forward (FF) Artificial Neural Network (ANN). It is a popular and old traditional machine learning model or algorithm that has been used to solve several problems in different human domains. MLP is made up of three layers - the input layer, output layer and hidden layer. The input dataset or variables to be processed are fed to the input layer. The inputted dataset are then fed into the hidden layer. The hidden layer contains several neurons which are the computational engine room of the algorithm; it performs the processing of the inputted dataset and feeds the output data or results to the output layer which gives the necessary prediction or classification value. This transmission of input data from the input layer through the hidden layer to the output layer is similar to a feed forward network in which the data flows direction is from input to the output layer. The neurons at the hidden layer in the MLP are thereby trained with the back propagation learning algorithm.

The neuron in the hidden layer sums up the total information that occurs in a MLP, including the bias.

$$\mathbf{y_0 = \sum_{i=1}^{n} w_i x_i + b} \quad (2)$$

y_0 is the linear form of the neurons. The non-linear form of the neurons is transformed when the activation function, ($\boldsymbol{f(x) = \frac{1}{1+e^{-x}}}$, is applied to equation (2), and this becomes

$$\mathbf{y_0 = f(x)\left|\sum_{i=1}^{n} w_i x_i + b\right|} \quad (3)$$

Where y_0=output, w_i = weight vector, x_i= scaled input vector, b=bias, f= transfer function and x=total sum of the weighted inputs.

Mathematically, the equation of MLP with several numbers of neurons is given as Hounmenou et al [9].:

$$\mathbf{y_0 = f\left|\sum WO_{kj}\left(\sum_{i=1}^{n} WI_{ij}x_i + b_1\right) + b_2\right|} \quad (4)$$

Where WI_{ij} = weight of the input layer, WO_{kj} = weight of the output layer, b_1=biased in the input layer and b_2= bias in the output layer.

MLPs are designed to solve problems that are linearly inseparable by using approximation method. The major applications of MLP ANN are in prediction and approximation, pattern classification and pattern recognition.

3.1.4 Support Vector Regressor (SVR)

Support Vector Regression (SVR) is a commonly used traditional supervised learning model or algorithm that is used to predict discrete values rather than classes of data. Support Vector Regression uses the same principle as the Support Vector Machines (SVMs). The basic working principle behind SVR is its ability to detect line of best fit. The best fit line in SVR is known as the hyperplane which has the maximum number of points.. SVR and SVM are similar but the basic difference between the two is that SVM is a classification model or algorithm which predicts classes of values while the SVR is used to predict real values rather a class of values. SVR recognizes the presence of non-linearity in a given dataset and this gives an efficient prediction model.

Mathematically, let Training dataset T, represented by

$$\boldsymbol{T = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}} \quad \mathbf{(5)}$$

Where $x \in X \subset R^n$ are the training inputs and $y \in Y \subset R$ are the training expected outputs.

A nonlinear function:

$$\mathbf{Y = } \boldsymbol{f(x) = \omega^T \Phi(x_i) + b} \quad (6)$$

where **ω** is the weight vector, **b** is the bias, and **Φ**(x_i) is the high dimensional feature space, which is linearly mapped from the input space x; the objective is to fit the training dataset T by finding a (x) that has the smallest possible deviation ε from the targets y_i.

3.1.5 Decision Tree Regression

Decision Trees (DT) is a very popular and common regression non-parametric supervised machine learning technique that has been around for a long time. It is a weak learner and suffers from bias and variance; decision trees with simple trees gives large bias and the one with complex trees gives large variance. The primary reason of using the DTs is to yield a predictive model for the values of the output variable, with the help of simple decision rules that have been derived from the essential features of the input predictor dataset or independent variables.

3.1.6 Ensemble Regression Models

Ensemble Regression models are very powerful and popular regression and classification techniques used in statistics and predictive machine learning problems. They combine individual multiple '*weaker learner*' models or several decision trees together to deliver a superior stronger prediction power. Ensemble regression models make use of special predictive techniques known as *Bagging*, *Boosting* and *Stacking* to reduce bias and variance in order to boost the accuracy of the target models. Examples of machine learning models or algorithms that makes use of ensemble regression include:-

- ***Random Forest (RF),***
- ***AdaBoost***
- ***Extra Trees and***
- ***XGBoost***

Example of an ensemble regression model that makes use of *bagging technique* for its predictive modeling is ***Random Forest*** *(RF);* it does not make use of boosting technique rather its trees are run in parallel. In boosting as the name implies, one is learning from other which in turn boosts the learning result. This process of boosting is sometimes referred to as *gradient boosting.* Boosting takes many forms such as *Adaptive Boosting* **(AdaBoost)**, Gradient Boosting **(Extra Trees)** and ***Extreme Gradient Boosting*** **(XGBoost).** ***Extra Randomized Trees or Extra Trees*** is an ensemble regression machine learning algorithm that combines the predictions from many decision trees to from a stronger prediction model by averaging predictions from many decision trees. Extra Trees has been reported to often achieve better prediction performance than the Random Forest (RF) algorithm; this is due to the fact that it uses simpler algorithm to construct the decision trees used as members of the whole ensemble.

Assume that the air pollution dataset is represented as

$$\mathbf{D = \{x_i, y_i): i=1,\ldots..n, x_i \in \mathbb{R}^m, y1 \in \mathbb{R}\}} \quad (7)$$

And selecting n observations with m features each and with a correspondingly variable y; then let $\check{Y}_i$ be defined as a result given by an ensemble represented by the generalized model:

$$\widehat{y_i} = \boldsymbol{\phi}(x_i) = \sum_{k=1}^{k} \boldsymbol{f_k}\,(x_i) \quad (8)$$

From the equation (8) , f_x is a regression tree, and $f_k(x_i)$ represents the score by the k_{th} tree to the i^{th} observation in the dataset.

Several ensemble algorithms can be used to create a hybrid ensemble machine learning models in order to improve prediction accuracy and performance by using several techniques such as *voting*, *averaging*, *stacking*, etc. For example XGBoost ensemble model can be combined with AdaBoost, Random Forest and Extra Trees to create a very powerful and highly accurate prediction model.

3.1.7 Lasso Regression Model

LASSO (Least Absolute Shrinkage and Selection Operation), is a regression machine learning algorithm used to reduce the problem of overfitting and improve accuracy level of the model. It uses shrinkage and regularization techniques to reduce or minimize the number of variables in a model.

3.2 Implementation Model

The implementation model for the proposed smart city air pollution modeling and prediction using various machine learning models is depicted in Fig.1. The Dataset was generated through the deployment of sensors in Internet-of-Things (IoTs) within a smart city and this historical dataset is used to train, model and predict the future occurrence of the air pollutants within the smart city.

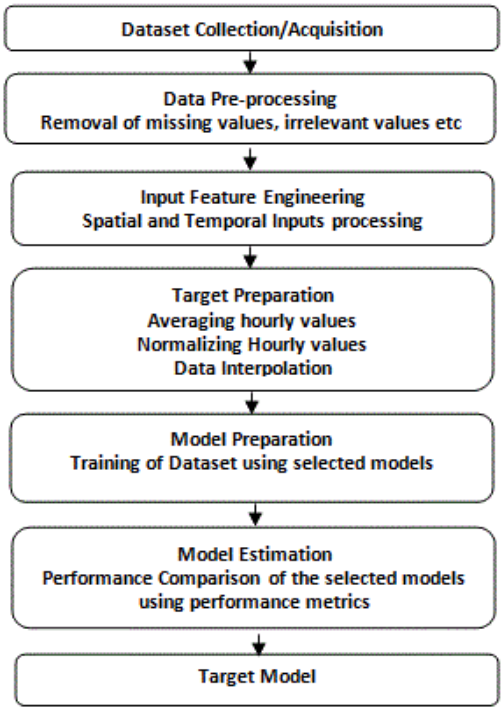

Fig.1. The architecture of the proposed Implementation air pollution modeling

3.3. Experimental Test-Bed

This paper was developed using mainly Python programming version 3 and python enabled Integrated Development Environment (IDE) popularly referred to as Anaconda Navigator and Jupyter Notebook; all these platforms are open source.

Equally, Machine Learning module for python such as Tensorflow, Scikitlearn and Keras were used to develop and program the predictive modeling algorithms used to model and predict the $PM_{2.5}$ concentrations in the smart city. Fig.2 shows the Anaconda Navigator management and development environment for python, Jupyter Notebook and other packages necessary to develop predictive algorithms for air pollution. Fig.3 depicts the Anaconda Tensorflow environment created to run the modeling and simulation for all the experiments. Fig.4 shows the screenshot of the Jupyter programming environment used throughout the running of the simulation and modeling programs for these experiments.

Python programs were developed in Python version 3 programming language. Python version 3 programming language contains inbuilt machine learning modules such as Tensorflow, Keras and Scikitlearn. These machine learning modules were used in carrying out the experiments and testing the models or algorithms in order to determine the best in performance.

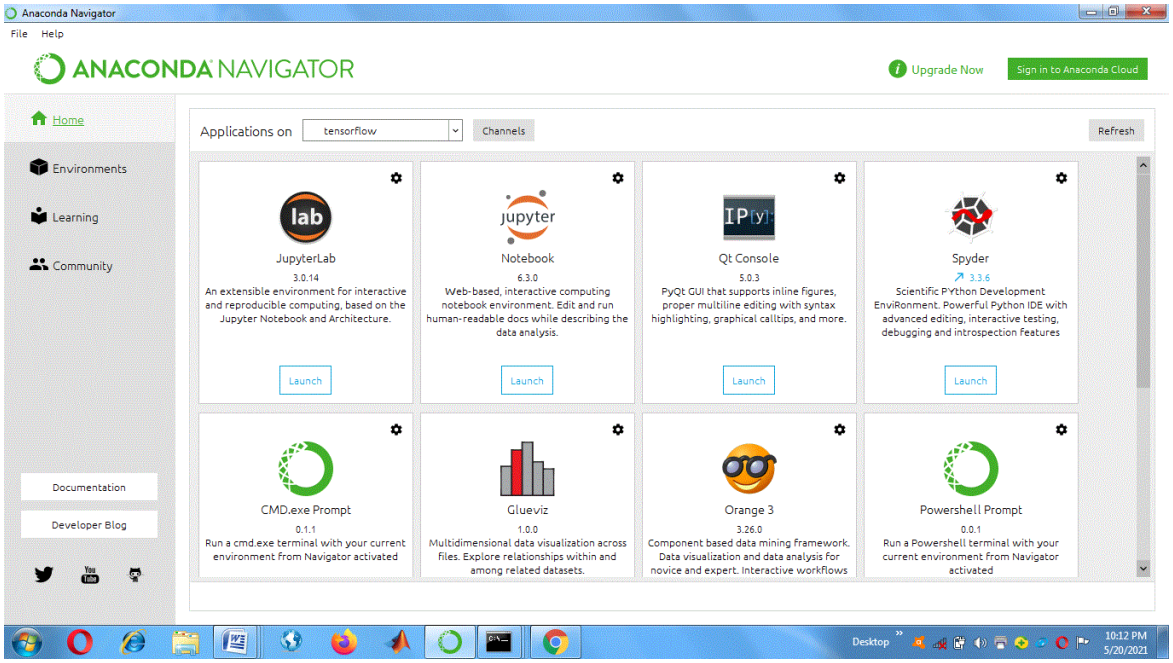

Fig.2. Anaconda Navigator Development Environment for Python

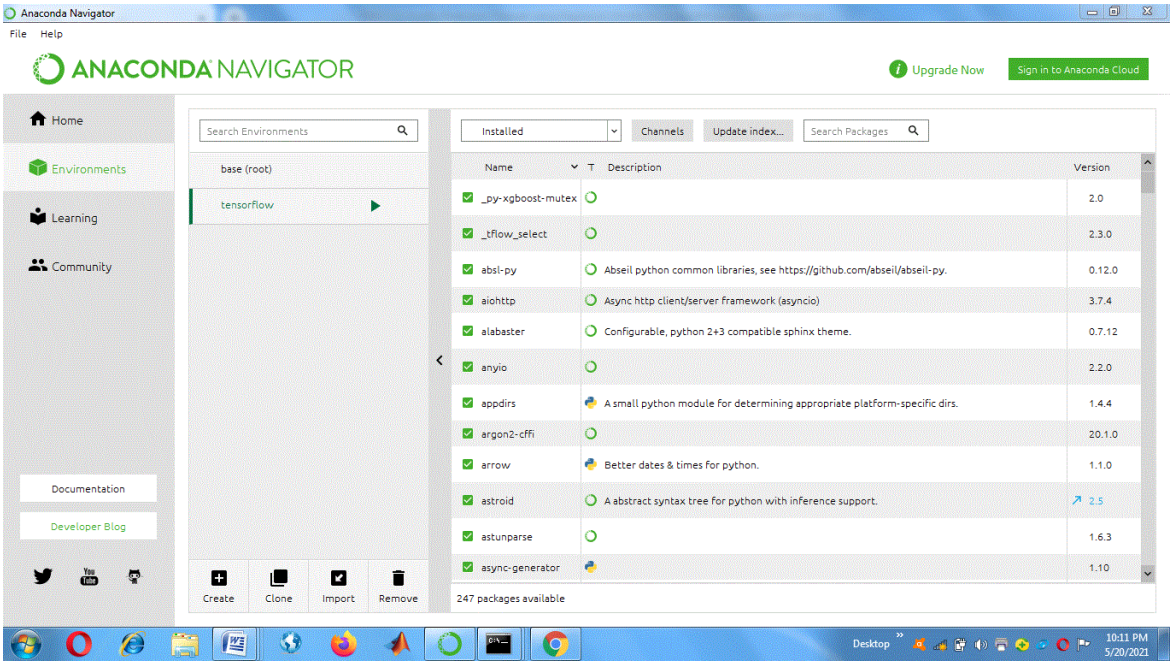

Fig.3. Anaconda Environment for Tensorflow Machine Learning Module for Python

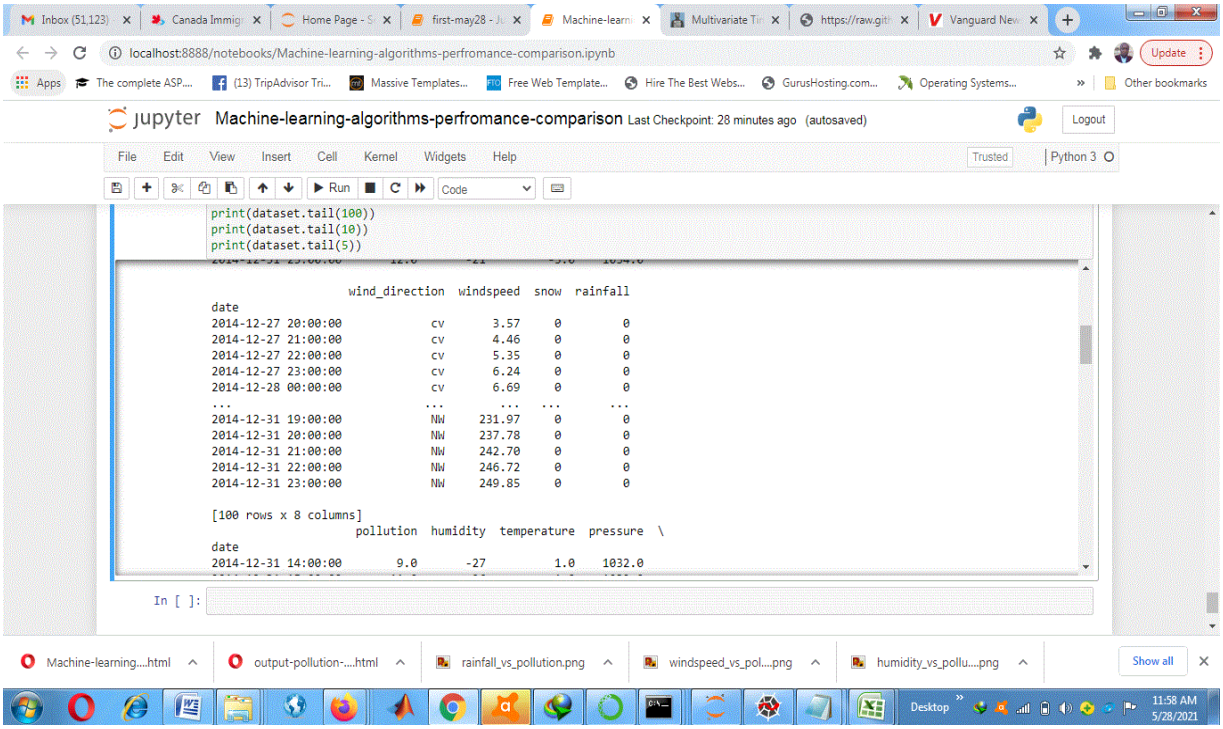

Fig. 4. A screenshot of Jupyter Notebook Integrated Development Environment (IDE) running Python 3 for Machine learning algorithms performance comparison

3.4. Data Collection and Pre-Processing

3.4.1 Data Collection and Data Pre-processing

The historical dataset of $PM_{2.5}$ air pollutants and other input predictor variables for five years (2010-2014) containing about 43,825 dataset samples of the city of Beijing, China (obtained online from US Embassy in Beijing, China website) was used to carry out the experiments. The input predictors to the models include the historical $PM_{2.5}$ pollutant concentrations, year, month of the year, day of the week, hour of the day and meteorological parameters comprising the following- air temperature, relative humidity, wind speed, wind direction, rainfall and snow. Data processing including data-preprocessing was carried out on the dataset in order to obtain a good dataset for modeling. During data-processing, all the null or missing values (NaN) (about 24 rows of the dataset) were dropped and replaced by the values adjacent to the rows that contained the NULL values using linear interpolation. The data was normalized by using the following formula (Zhao et al.[10]) :

$$Z_i = \frac{x_i - min(x)}{max(x) - min(x)} \quad (9)$$

where $x=(x_1, \ldots\ldots x_n)$ and Z_i is the normalized data.

In order to improve the performance accuracy of the prediction models, a time-series of 6-hour, 12-hour and 24-hour averages for $PM_{2.5}$ concentrations were introduced and performed during feature engineering stage of the experiment.

The dataset was split into 80% for training and 20% for testing and validation.

3.4.2 Input Predictor Variables

The input predictor or independent variables used in this experiment include the following:

1. *pm2.5 (pollutant concentration),*
2. *hour,*
3. *day (Day of the week),*
4. *month (Month of the year),*
5. *temperature (temperature of the atmosphere),*
6. *pressure (Air pressure),*
7. *humidity (Relative humidity of the atmosphere),*
8. *windspeed*
9. *Is (Accumulated snow), and*
10. *Ir (Accumulated rainfall or precipitation).*

Wind direction (wind_direction) was dropped during data pre-processing and feature engineering processes.

3.4.3 Output Variable

The output variable or dependent variable to be predicted is the pollutant parameter particulate matter $PM_{2.5}$.

3.5 Performance Evaluation Metrics

In order to determine or evaluate the best machine learning Air pollution prediction models quantitatively in terms of error bands or the prediction accuracy, the following statistical performance metrics- Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of Determination or Variance (R^2) were employed and calculated as shown in Eqs. (10)-(12).

A. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - M_i| \quad (10)$$

B. Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|P_i - M_i|^2} \quad (11)$$

C. Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum(M_i - P_i)^2}{\sum(M_i - \overline{M_i})^2} \quad (12)$$

where n is the number of data in the test dataset,P_i and $M_i$are the predicted and measure value for the i^{th} hour and $\overline{M_i}$ is the mean of all the measured values for the i^{th} hour. The higher the value of R^2, the more accurate and better the prediction result while the lower the values of RMSE and MAE, the higher the accuracy of the prediction model or algorithm.

IV. RESULTS AND DISCUSSION

4.1 Results

Figs. 5-13 show the scatterplots regression results between predicted and measured values for Linear Regression (MLR), MLP-ANN, SVM Regressor, Lasso Regressor, Extra Trees Regressor, Decision Tree, Decision Tree with AdaBoost and XGBoost Regressor respectively.

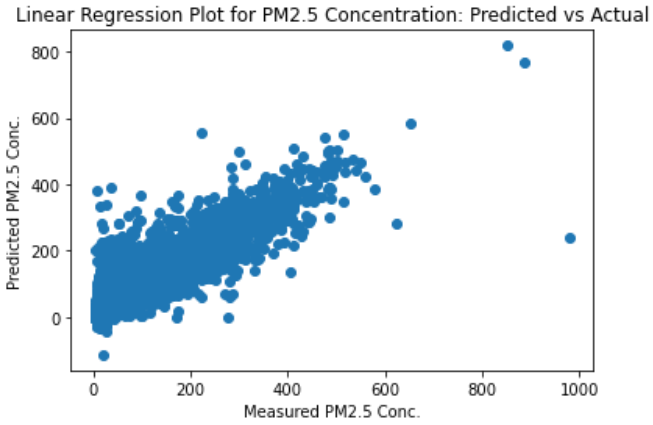

Fig.5. Multi Linear Regression Scatterplot: Predicted versus Measured

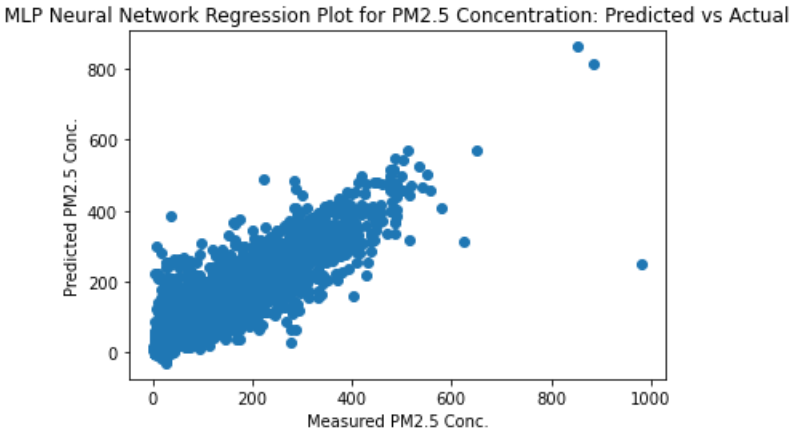

Fig. 6 MLP Artificial Neural Network (ANN) Regression scatterplot: Predicted versus Measured

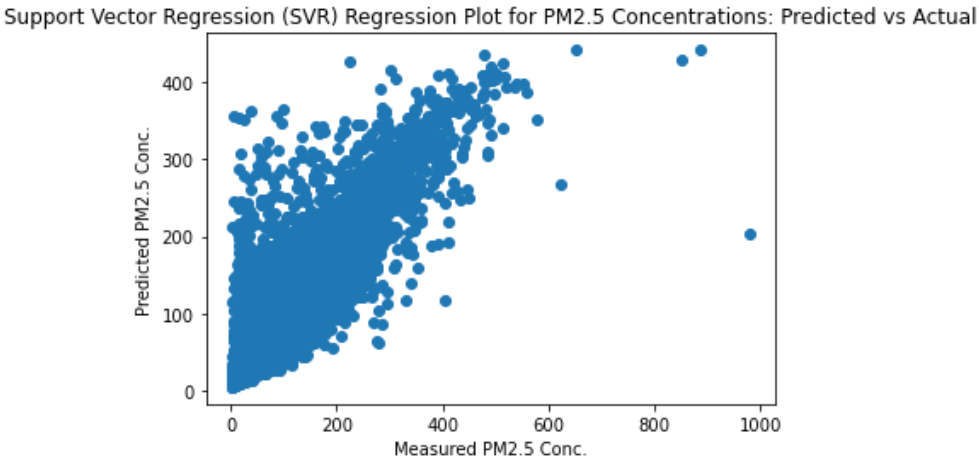

Fig.7 . SVR Regression Scatterplot: Predicted versus Measured

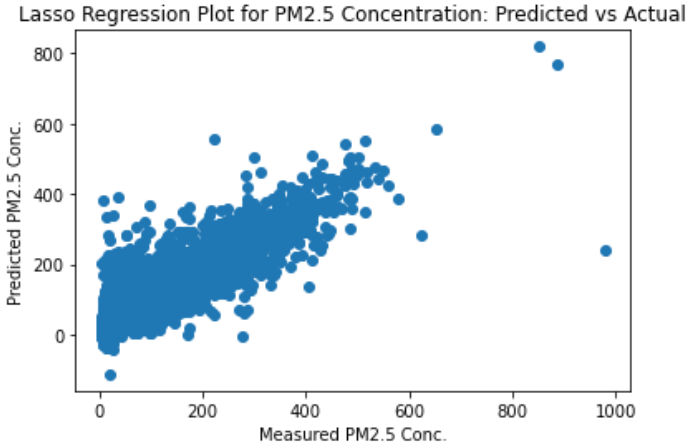

Fig. 8. Lasso Regression Algorithm Scatterplot: Predicted versus Measured

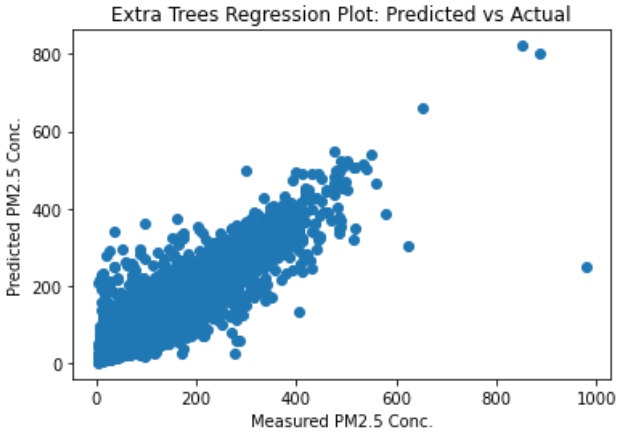

Fig. 9. Extra Tree Algorithm: Predicted versus Measured

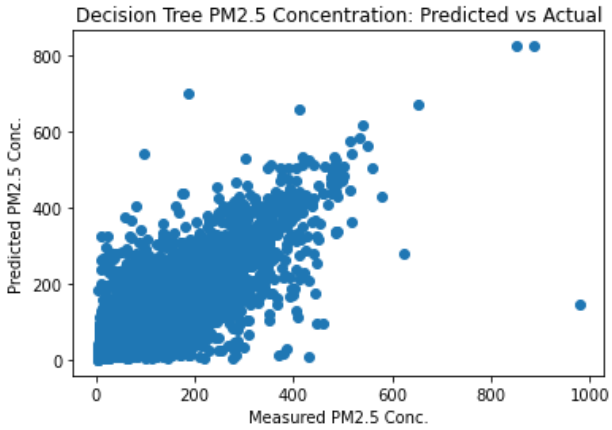

Fig.10. Decision Tree Algorithm: Predicted versus Measured

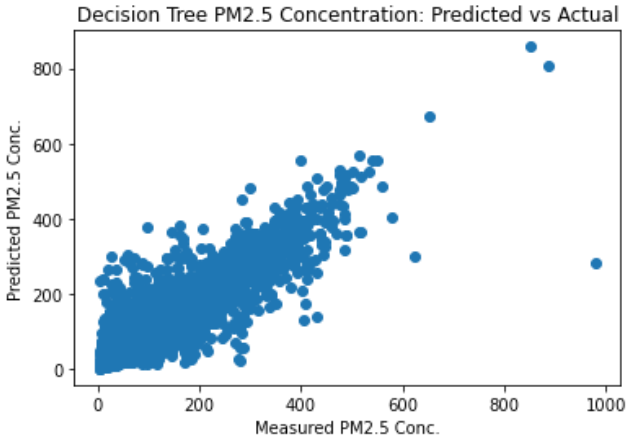

Fig.11. Boosted Decision Tree (Decision Tree boosted with AdaBoost Algorithm): Predicted versus Measured

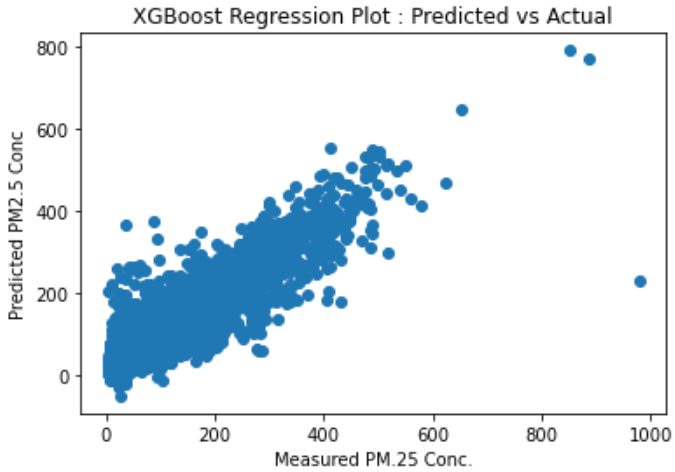

Fig.12. XGBoost Algorithm Scatterplot: Predicted versus Measured

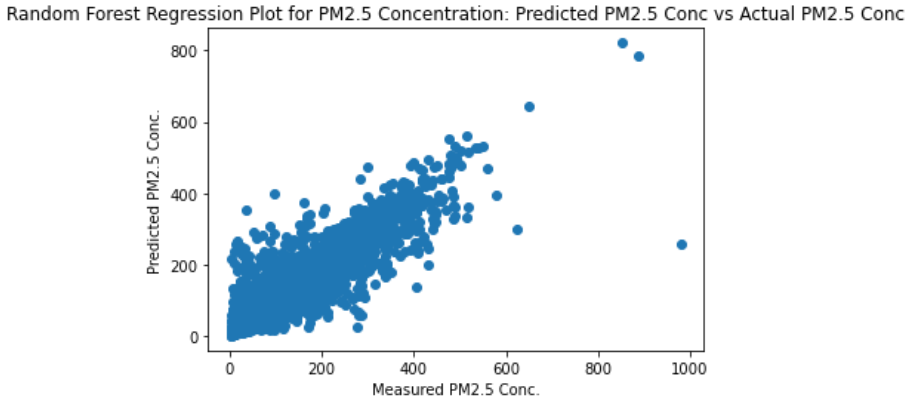

Fig. 13. Random Forest Scatterplot: Predicted versus Measured

Fig. 14 shows the screenshot for the Accuracy or R^2 score for the nine (9) Machine Learning Regression algorithms computed with Python program in Jupyter Notebook IDE.

```
print("Boosted Decision Trees (Decision Tree++AdaBoost score: ", boosted_tree_score)
print("XGBoost score:", xgb_score)
print("\n")

Accuracy Scores:
Linear regression score:  0.8118593972485936
Neural network regression score:  0.8378114213897029
Support Vector Regression score :  0.7674232232775885
Lasso regression score:  0.8117453954431458
Random Forest score:  0.8473577929983932
Extra Trees score:  0.8515674203500716
Decision Trees score:  0.683464826712725
Boosted Decision Trees (Decision Tree++AdaBoost score:  0.8497335826456425
XGBoost score: 0.8525459110700417

In [ ]:
```

Fig. 14. Screenshot of Accuracy or R^2 score of the Nine (9) machine learning regression models in Jupyter Notebook Python program

Table 2. The comparison of results of machine learning algorithms obtained from experimental runs

S/N	Machine learning Regression Model	R^2 or Accuracy Score ($\mu g/m^3$)	MAE Score ($\mu g/m^3$)	RMSE score ($\mu g/m^3$)
1	Multi Linear Regression	0.812	23.76	38.58
2	MLP Neural Network	0.838	21.97	35.82
3	SVR	0.767	25.76	42.89
4	Lasso	0.812	23.71	38.59
5	Extra Trees	0.852	20.37	34.26
6	Random Forest	0.847	20.67	34.75
7	Decision Trees	0.683	30.03	50.04
8	Decision Trees + AdaBoost	0.850	10.68	34.48
9	XGBoost	0.853	21.02	34.15

Figs. 15-17 depict the barplots showing the comparative performance results (R^2, MAE and RMSE) respectively of the nine machine learning algorithms obtained from the experiments.

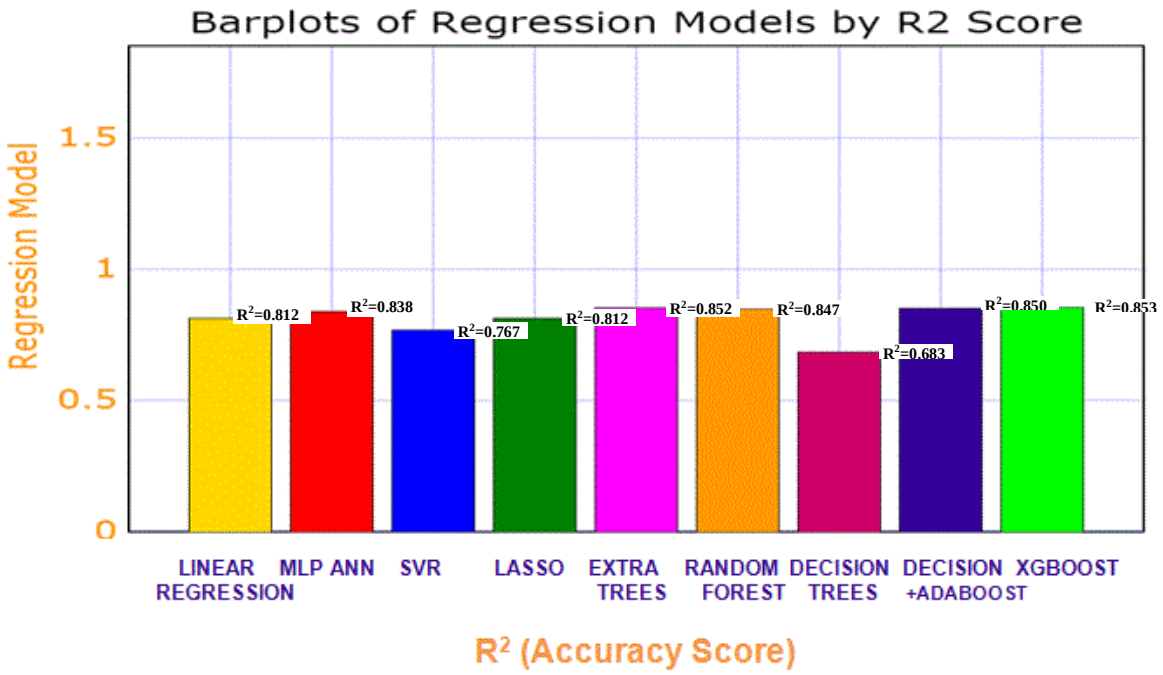

Fig. 15. The performance Evaluation result for the machine learning regression algorithms for PM2.5 Prediction using R^2 metric

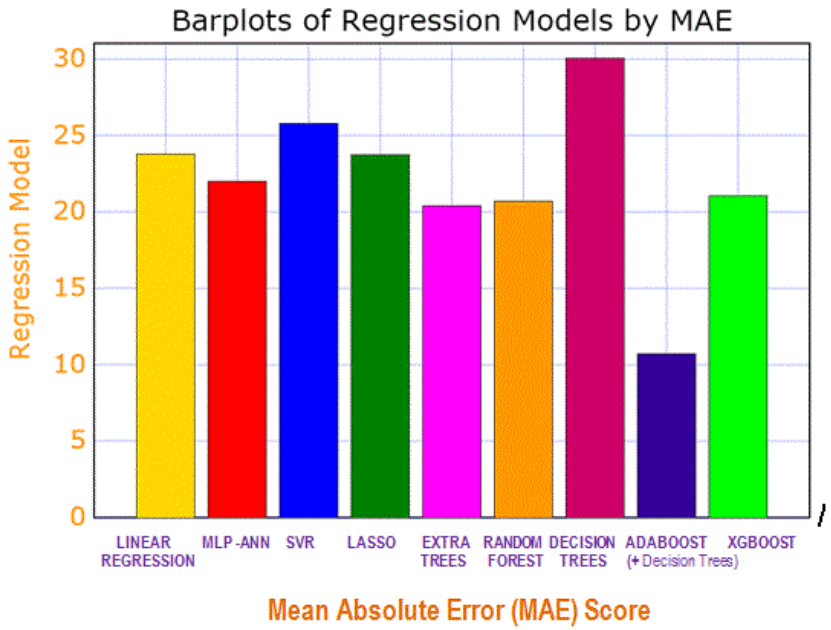

Fig. 16. The performance Evaluation result for the machine learning regression algorithms for PM2.5 Prediction using Mean Absolute Error (MAE) metric

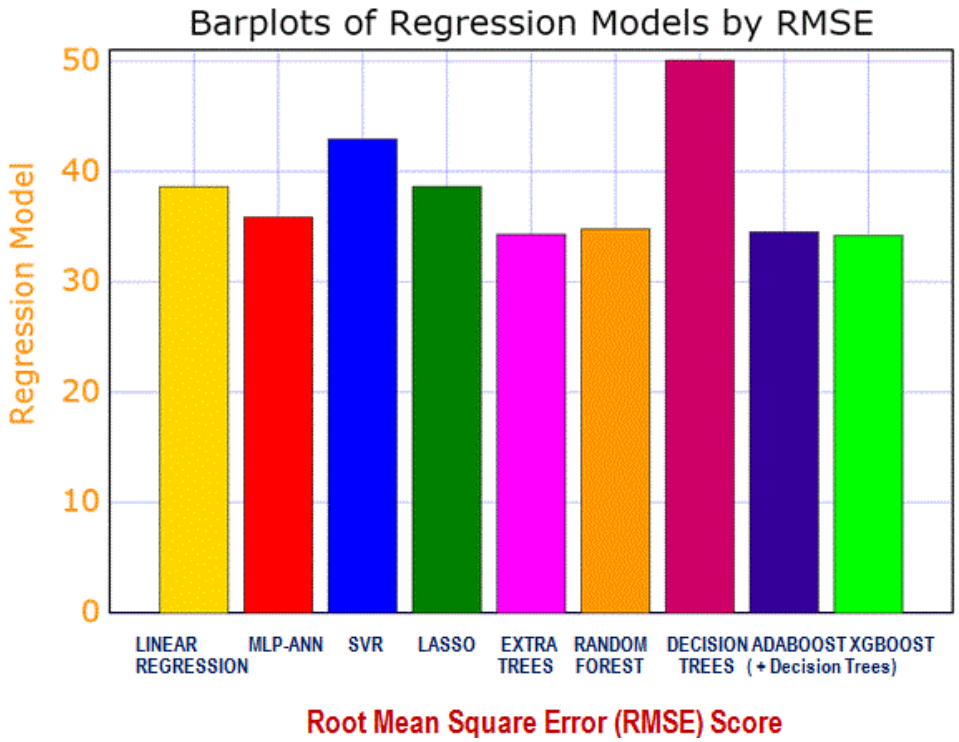

Fig. 17. The performance Evaluation result for the machine learning regression algorithms for $PM_{2.5}$ Prediction using Root Mean Square Error (RMSE) metric

4.2 Discussion Of Results

Results of all the eight regression models on the smart city historical dataset were presented on Table 2 and Figs.15-17.

From Table 2 and Figs 15-17 the results of the performance comparisons of the nine supervised learning regression models show that ensemble algorithms of XGBoost, Decision Trees + AdaBoost, Random Forest, Extra Trees, performed relatively better in accuracy than the traditional machine learning models such as ordinary Decision Tree (DT), SVR and MLP ANN as expected because ensemble regression and classification algorithms perform better than the traditional single algorithms. This is because ensemble algorithms combine the weaker learner models together into stronger and more accurate algorithms or models. From the results, it can be seen that XGBoost model scored the highest R^2 score of 0.853 followed by Extra Trees (R^2=0.852), Boosted Decision Tree (Decision Trees +AdaBoost) came 3rd with R^2=0.850), then followed by Random Forest (R^2=0.847). Amongst the traditional machine learning algorithms, MLP Neural Network (MLP ANN) came 5th with R^2=0.838, followed by Lasso algorithm and Linear Regression (MLR) both having R^2=0.812, followed by SVR with R^2=0.767 and Decision Trees came last with R^2=0.683 in accuracy of prediction. A traditional machine learning algorithm such as Decision Trees prediction accuracy can be boosted or improved by boosting it with an ensemble algorithm such as AdaBoost, XGBoost, Random Forest or Extra Trees.

In terms of errors in prediction using MAE as expected the ensemble learning algorithms have the lowest MAE starting with (Decision Trees+AdaBoost) (MAE=10.68) followed by Extra Trees (MAE=20.37), Random Forest (MAE=20.67), XGBoost (MAE=21.12), the traditional single machine learning algorithms score the lowest in MAE with MLP (MAE=21.97), Lasso (MAE=23.71), Linear Regression (MAE=23.76), and traditional Decision Trees algorithm (MAE= 30.03).

Similarly, evaluating the performances of the machine learning algorithms using RMSE, it was observed from Table 2 that the experimental results showed that the ensemble algorithms scored first with the lowest RMSE with XGBoost (RMSE=34.15μ/m^3), Extra Trees came second with RMSE=34.26 μ/m^3, followed by Boosted Decision Trees (Decision Trees + AdaBoost) with RMSE=34.48 μ/m^3 . Another ensemble algorithm

Random Forest scored fourth with RMSE=34.75μ/m^3 while the best traditional machine learning algorithm, MLP Neural Network (MLP-ANN) scored fifth with RMSE=35.82 μ/m^3 while Lasso algorithm scored sixth with RMSE=34.59 μ/m^3. Traditionally weak learner algorithm Decision Tree scored last with the highest RMSE value of 50.04 μ/m^3.

These results show that ensemble models with gradient descent boosting and bagging techniques produce prediction results with higher accuracy and lower prediction errors.

Figs. 5-13 show the regression scatterplots of the observed value versus the measured values for the eight machine learning models. The scatterplots show that there is correlation between the predicted and measured values of the predictors.

V. CONCLUSION

Smart city solutions can be deployed in the management of environmental pollution and can be deployed for air quality prediction in order to forecast the concentrations of air pollutant parameters present in a city or metropolis. This is important so as to alert city administrators and the general public to know the implications of the environment in order to protect their health. Machine learning is a branch of Artificial Intelligence that can be used to predict or forecast the possible concentrations of air pollutants in the atmosphere before it occurs using past historical dataset of air pollutants and meteorological parameters of the same locality. This paper has demonstrated that it is possible to use machine learning algorithms to model and predict air quality of a city or metropolis.

In this work, particulate matter PM2.5 predictions were carried out using nine (9) different machine learning models. The prediction performances of these machine learning models were evaluated using statistical performance metrics such as MAE, RMSE and R2. The results showed that out of the nine machine learning models such as ensemble learning algorithms: XGBoost, Extra Trees, AdaBoost boosted Decision Trees, Random Forest in that order performed better in terms of prediction accuracy and reduced prediction errors compared to the traditional machine learning algorithms such as Linear Regression, SVR, MLP Neural Network, Lasso. This is expected because ensemble learning algorithms combine several 'weaker' learning algorithms such as decision trees into stronger and more accurate prediction models. Therefore, it can be deduced that a traditional machine learning algorithm such as Decision Trees can be boosted in terms of accuracy of prediction by combining it with an ensemble learning model such as AdaBoost or Random Forest or Extra Trees or XGBoost.

The experimental results equally showed weather or meteorological parameters in an area or location such as temperature, air pressure, windspeed, relative humidity, humidity, rainfall or precipitation etc. have direct correlation or effects on the hourly concentrations of PM2.5 in that location or area.

For future work , it is recommended that the performance evaluation of different classification algorithms for PM2.5 prediction and also to improve on the prediction accuracy of Air pollution models by combining several high-performing ensemble learning models such XGBoost, AdaBoost, Extra Trees, Random Forest together to create a very powerful high performing and accurate prediction model.

REFERENCES

[1]. Castelli, Mauro; Clemente, Fabiana Martins; Popovic, Ales ; Silva, Sara and Vanneschi, Leonardo (2020). A Machine Learning Approach to Predict Air Quality in California. Hindawi Complexity Volume 2020, Article ID 8049504, https://doi.org/10.1155/2020/8049504

[2]. Mallet, Vivien and Spotisse, Bruno (2008). *Air* Quality modeling: From deterministic to stochastic approaches. Computers & Mathematics with Applications. Volume 55, Issue 10, May 2008. pp. 2329-2337.

[3]. Džaferovic Emina and Kanita K. Hadžiabdic (2021). "Air Quality Prediction and Application using Machine Learning methods: A case study of Bjelave Neighborhood". Technological Systems and Applications. Lecture Notes in Network Systems 142.

[4]. Aceves-Fernández, M.A., Domínguez-Guevara, R., Pedraza-Ortega, J.C. and Vargas-Soto, J.E. (2020). *Evaluation of Key Parameters Using Deep Convolutional Neural Networks for Airborne Pollution (PM10) Prediction*. Hindawi. Discrete Dynamics in Nature and Society, Volume 2020, February 2020. PP 1-14. https://doi.org/10.1155/2020/2792481

[5]. Pan Bingyne (2017). "Application of XGBoost algorithm in hourly PM2.5 Concentration prediction". IOP Conference Series: Earth and Environmental Science 113 (2018) 012127 doi :10.1088/1755-1315/113/1/012127

[6]. Kıymet Kaya & Şule GündüzÖğüdücü (2020). "Deep flexible Sequential (DfS) Model for Air pollution forecasting". Scientific Reports (2020) 10:3346 | https://doi.org/10.1038/s41598-020-60102-6

[7]. Joharestani; Medhi Zamani; Chunxiang C.; Xiliang N.; Barjeece B. and Somayeh T. (2019). *$PM_{2.5}$ Prediction Based on Random Forest, XGBoost and Deep Learning Using Multisource Remote Sensing Data*. .Atmosphere2019,10, 373; doi:10.3390/atmos10070373

[8]. Hahez Adam (2021). Multiple Linear Regressions (MLR). Investopaedia.com. Accessed online at https://www.investopedia.com/terms/m/mlr.asp on September 30, 2021.

[9]. Hounmenou, Castro Gbêmêmali ; Gneyou, Kossi Essona and Kakaï, Romain Glèlè (2021). *A Formalism of the General Mathematical Expression of Multilayer Perceptron Neural Networks.* Preprints. May 18, 2021. doi:10.20944/preprints202105.0412.v1

[10]. Zhao, N., Wen, X., Yang, J., Li, S., and Wang, Z. (2015). Modeling and forecasting of viscosity of water-based nanofluids by radial basis function neural networks. Powder Technology, 281:173183.

Fidelis C. Obodoeze, et. al. "Comparative Evaluation of Machine Learning Regression Algorithms for PM2.5 Monitoring."*American Journal of Engineering Research (AJER)*, vol. 10(12), 2021, pp. 19-33.