PREDICTING AIR QUALITY(PM2.5) LEVELS USING LINEAR REGRESSION BASED RECURSIVE FEATURE ELIMINATION WITH RANDOM FOREST REGRESSION(RFERF)

Report submitted to the SASTRA Deemed to be University
in partial fulfilment of the requirements
for the award of the degree of

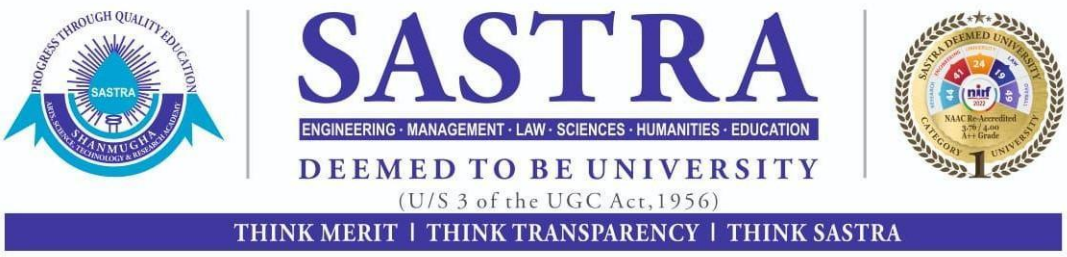
B.Tech., Computer Science and Engineering

Submitted by

MALLINENI MEGHANA (223003063)

MANNEM ESWARLAKSHMI (223003064)

NABBU SHAIK ANEESA(223003068)



Department of Computer Science and Engineering

SRINIVASA RAMANUJAN CENTRE

Kumbakonam, Tamil Nadu, INDIA - 612 001

JUNE 2023

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

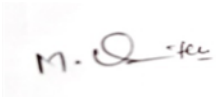
Department of Computer Science and Engineering

SRINIVASA RAMANUJAN CENTRE

Kumbakonam, Tamil Nadu, INDIA - 612 001

Bonafide Certificate

This is to certify that the project report titled " **Predicting Air Quality(PM2.5) levels using Linear Regression based Recursive Feature Elimination with Random Forest Regression(RFERF) "** ', submitted in partial fulfilment of the requirements for the award of the degree of B.Tech.,Computer Science and Engineering to the SASTRA Deemed to be University, is a bonafide record of the work done **Ms. MALLINENI MEGHANA(Reg. No. 222003063, B. Tech-CSE), Ms. MANNEM ESWARLAKSHMI (Reg. No. 223003064, B. Tech-CSE), Ms. NABBU SHAIK ANEESA (Reg. No. 222003068, B. Tech-CSE)** during the final semester of the academic year 2022-2023, in the Srinivasa Ramanujan Centre, under my supervision. This project report has not formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title to any candidate of any University.

Signature of the Project Supervisor: 

Name with Affiliation : **Mrs.VANITHA M**

Assistant Professor-II/CSE/SRC.

Date : 13-June-2023

Project *Viva Voce* held on_____

Examiner I **Examiner II**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

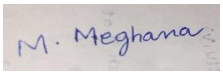
Department of Computer Science and Engineering

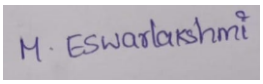
SRINIVASA RAMANUJAN CENTRE

Kumbakonam - 612 001

Declaration

We declare that the project report titled **" Predicting Air Quality(PM2.5) levels using Linear Regression based Recursive Feature Elimination with Random Forest Regression (RFERF) "** was submitted by us in an original work done by us under the guidance **Mrs. Vanitha. M**, **Assistant Professor-II, Department of CSE, SRC, SASTRA Deemed to be University, Kumbakonam,** during the final semester of the academic year 2022-23, in the **Srinivasa Ramanujan Centre**. The work is original and wherever we have used the materials from other sources, we have given due credit and cited them in the text of the project report. This project report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidates : 1. 

2. 

3. 

Name of the candidates : 1. MALLINENI MEGHANA (223003063)

2. MANNEM ESWARLAKSHMI (223003064)

3. NABBU SHAIK ANEESA (223003068)

Date : 13-June-2023

Acknowledgements

We pay our sincere obeisance to the God Almighty for his grace and infinite mercy and for showering on us his choicest blessings.

We would like to thank our honourable Chancellor **Prof. R. Sethuraman,** Vice-Chancellor **Dr. S. Vaidhyasubramaniam** and **Dr. S. Swaminathan**, Dean - Planning & development for the encouragement and strategic support at every step of our college life.

We extend our sincere thanks to **Dr. R. Chandramouli,** Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

We extend our heartfelt thanks to **Dr. V. Ramaswamy**, Dean, Srinivasa Ramanujan Centre, and **Dr. A. Alli Rani**, Associate Dean, Srinivasa Ramanujan Centre.

We would like to express our deep sense of gratitude to our guide **Smt. M . Vanitha**, Assistant Professor-II, CSE Srinivasa Ramanujan Centre, who was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped us in making progress throughout our project work.

We would like to thank our panel members **Smt. S .Priyanga** and **Shri. Eashwar K B,** for their valuable comments and insights which made this project better.

We would like to place on record the benevolent approach and painstaking efforts of guidance and correction of **Smt. S. Hemamalini** and **Dr. K. Manjula**, the project coordinators and all the department staff members to whom we owe our hearty thanks forever.

We gratefully acknowledge all the contributions and encouragement from our family members and friends resulting in the successful completion of this project. We thank you all for providing us an opportunity to showcase our skills through this project.

We also dedicate this work to all our well-wishers, with love and affections.

List of Figures

List of Tables

Abbreviations

RFERF	Recursive Feature Elimination with Random Forest Regression
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MedAE	Median Absolute Error
MAPE	Mean Absolute Percentage Error
MSLE	Mean Squared Log Error
EV	Explained Variance Score
PM	Particulate Matter

ABSTRACT

Air pollution has become a rising and concerning issue at the same time. In many cities across India, air pollution levels are beyond the maximum and if this trend continues, air pollution can lead to many physiological and ecological disasters.The major pollutants SO2,NO2,PM2.5,PM10.But we are going to focus on PM2.5, as this pollutant is very tiny in size that it is almost microscopic.PM2.5 refers to Particulate Matter(which is a combination of solid particles and liquid droplets) whose size is less than 2.5 micro meters.

In this model we predict PM2.5 concentration in the air using Linear Regression based Recursive Feature Elimination with Random Forest Regression(RFERF).Three models namely Decision Tree, Random Forest Regression and Gradient boosting Regression are used and the best one with optimal features and better accuracy is selected. All the models are compared with different metrics and are visualized through graphical representations.

Table of Contents

CHAPTER 1

SUMMARY OF THE BASE PAPER

Title: Spatial Air Quality Index and Air Pollutant Concentration prediction using Linear Regression based Recursive Feature Elimination with Random Forest Regression (RFERF): a case study in India

Journal name: Measurement

Publisher: S. Roopashree, J. Anitha, T.R. Mahesh, V. Vinoth Kumar, Wattana Viriyasitavat, Amardeep Kaur

Year: 2022

1.1 Introduction:

In this paper, the authors present a model that uses machine learning to predict PM2.5 concentration levels in India. Specifically, they use the linear regression-based recursive feature removal and random forest regression (RFERF) model and compare it to other good machine learning models.

Air pollution has become a major environmental problem in India, especially in cities like Delhi where population and energy consumption are increasing rapidly. This article highlights the importance of air quality forecasting to warn people of good air quality and reduce the overall impact of poor air quality on human health.

Overall, this article presents a comprehensive approach to predicting air quality in India using machine learning and highlights the urgent need to address air pollution. The purpose of this system is to create a model to predict PM2.5 concentration in air less than 2.5 microns using iterative feature removal based on linear regression with random forest regression (RFERF) and gradient boost regression.

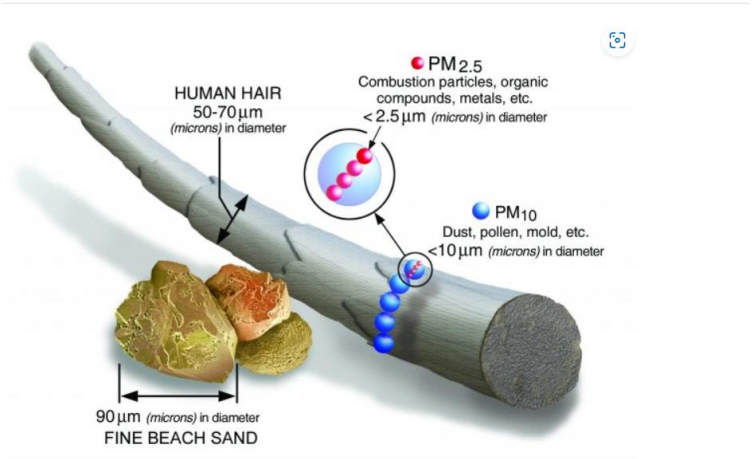


Fig 1 :Demonstration and Comparison of PM2.5 with other particles

1.2 proposed System:

The proposed system is a machine learning model for predicting PM2.5 concentration present in the air. It uses a Linear Regression based Recursive Feature Elimination with Random Forest Regression(RFERF) algorithm and also Gradient Boosting Regressor. RFERF basically includes data cleaning, recursive feature elimination ,hyper-tuned Random Forest Regression, and Performance evaluation using different parameters like MSE,MAE,MSLE,RMSE etc. The system chooses the best optimal features and performs the fitting.

1.3 Dataset:

This database contains hourly air pollution data from 12 countries that govern air pollution. Air quality data from the Beijing Environmental Monitoring Center. The weather data for each weather station is similar to the closest one from the China Meteorological Administration. The period is from March 1, 2013 to February 28, 2017. Missing data were reported by NA.

This dataset which contains 35064 observations is splitted in 80:20 ratio i.e. 80% training data and 20% testing data

NAME OF THE FACTOR	DISCRIPTION	UNIT
NO	Row number	
YEAR	Year of data in this row	
MONTH	Month of data in this row	
DAY	Day of data in this row	
HOUR	Hour of data in row	
PM2.5	PM2.5 concentration	(ug/m^3)
PM10	PM10 concentration	(ug/m^3)
SO2	SO2 concentration	(ug/m^3)
NO2	NO2 concentration	(ug/m^3)
CO	CO concentration	(ug/m^3)
O3	O3 concentration	(ug/m^3)
TEMP	temperature	Degree celsius
PRES	pressure	(hPa)
DEWP	Dew point temperature	Degree celsius
RAIN	precipitation	(mm)
WD	Wind direction	
WSPM	Wind speed	(m/s)

Table 1

1.4 Methodology:

1.4.1 Random Forest Regressor:

Random forest regressor is a machine learning algorithm used for regression problems. It is an ensemble learning algorithm that combines multiple decision trees to make a more accurate prediction. It is called a random forest because it consists of a large number of decision trees that work together to give a more robust result.

Each tree in the random forest is constructed based on a random subset of the training data, and a random subset of the input variables. This helps prevent overfitting, which can occur when the algorithm becomes too complex and fits itself too well to the training data.

To predict the outcome, each decision tree in the forest is used to make a prediction. The overall prediction is then decided based on the average prediction of all the trees in the forest. This approach produces a more accurate and reliable prediction as compared to a single decision tree.

Random forest regressor has several advantages over other regression algorithms. One advantage is that it can handle a large number of input variables and can effectively deal with missing data. It is also very quick to train, which makes it an ideal algorithm for handling large datasets.

Another advantage of using a random forest regressor is that it can handle non-linear and complex relationships between the input variables and the output variable. It is able to detect these relationships even when they are not obvious in the data.

In conclusion, the random forest regressor is a powerful machine learning algorithm that can be used to accurately predict the outcome of a regression problem. It is fast, robust, and able to handle complex relationships between variables

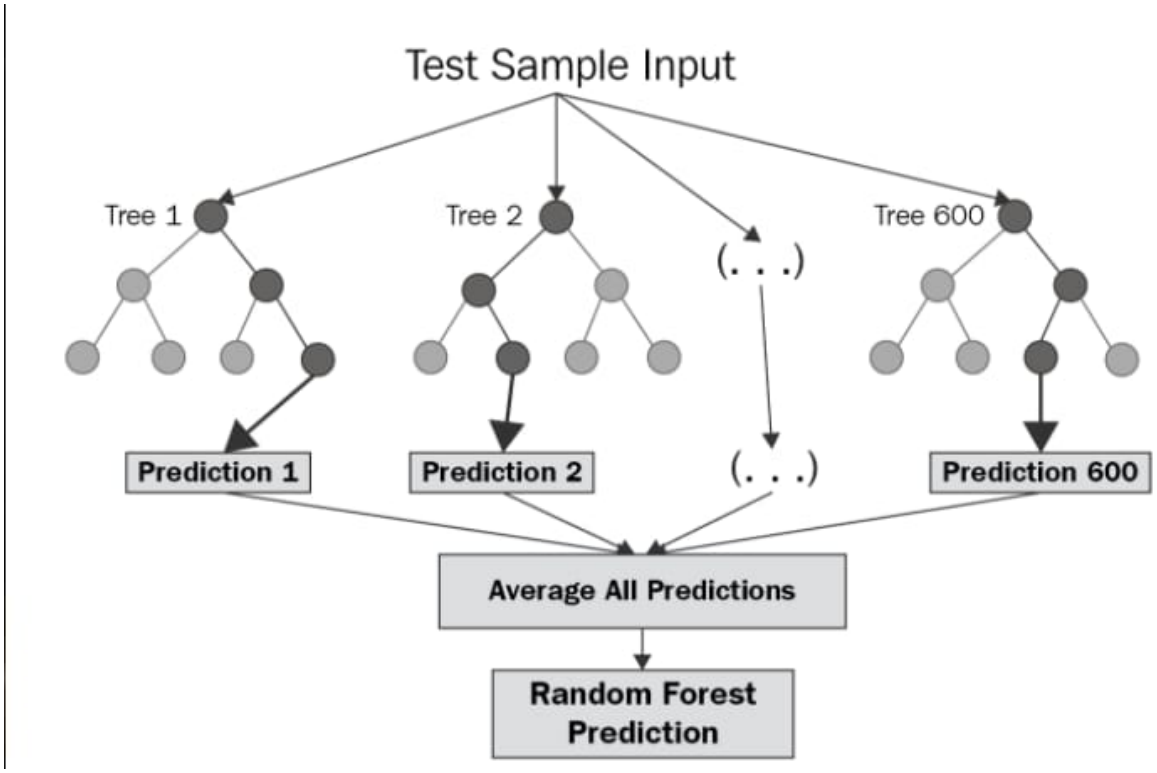


Fig 2 : Demonstration of Random Forest Regressor

1.4.2 Decision Tree Regressor:

Decision tree regressor is a machine learning algorithm that is used for regression problems. It is a type of decision tree algorithm that is used to solve decision-making problems where the outcome is a continuous numerical value, rather than a categorical value.

The algorithm forms a tree-like model of decisions and their possible consequences and is based on a hierarchical structure of nodes. The tree starts with a root node and branches out into different decision paths, eventually resulting in leaf nodes. These leaf nodes represent the final predictions for the input data.

In decision tree regressor, the algorithm creates split points based on the best value of impurity metric. The metric used can be mean squared error, mean absolute error, or any other dissimilarity metric. It takes the form of a tree structure in which each node represents a feature and the branches represent either conditional relationships or possible output numerical values.

At each node, the algorithm searches for the best value to split the data based on the chosen metric. The process continues until a stopping criterion is met, such as a maximum depth being reached or a minimum number of instances being present.

One of the advantages of using decision tree regressor is that it is easy to understand and interpret, and can handle any data type. It is also a fast algorithm and can handle data with missing values.

However, one of the potential drawbacks of decision tree regressor is that it is prone to overfitting. Overfitting occurs when the algorithm creates too many splits that are tailored specifically to the training data, which creates poor results when new data is introduced.

In summary, decision tree regressor algorithm is a popular and widely used algorithm for solving regression problems. It is simple and easy to interpret, and can handle different types of data. However, it is important to be aware of its limitations, specifically its tendency to overfit, and take steps to prevent it

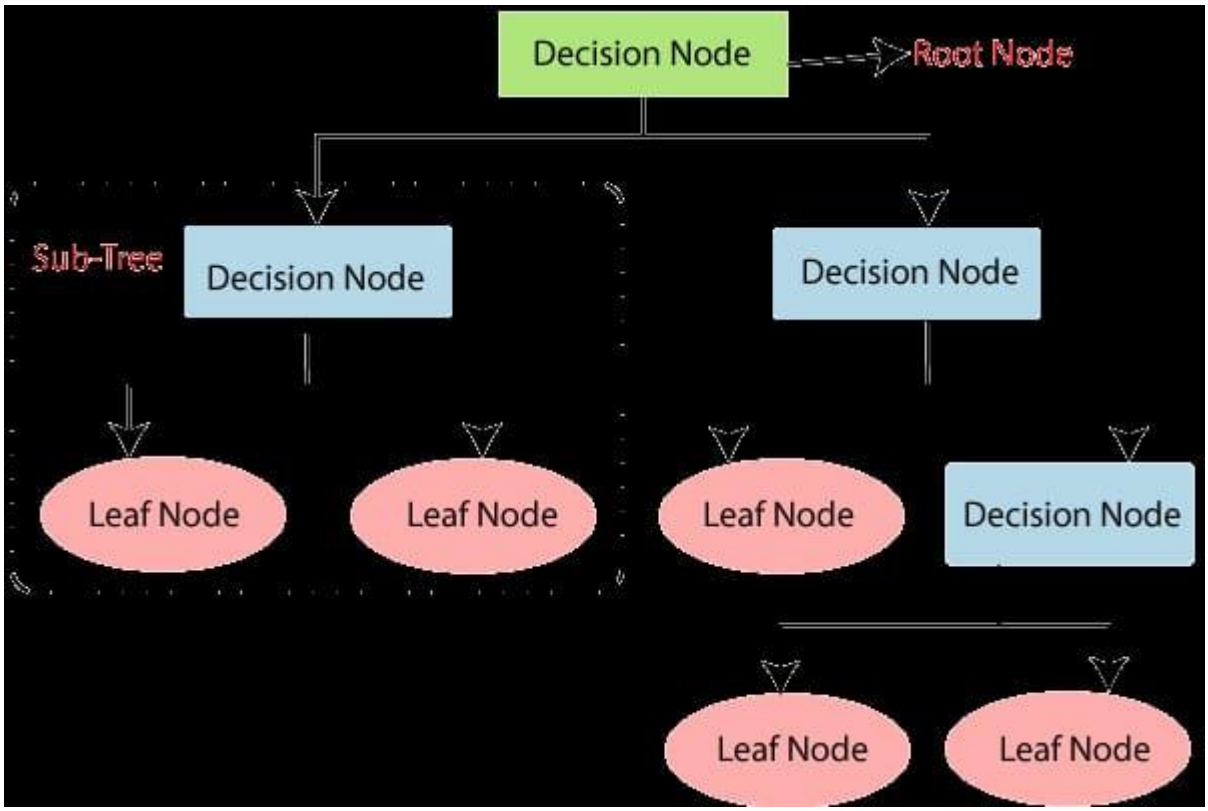


Fig 3 : Demonstration of Decision Tree Regressor

1.4.3 Gradient Boosting Algorithm:

Gradient Boosting Regressor Algorithm is a supervised machine learning algorithm used for regression problems. It belongs to the boosting family of algorithms and is an improvement over the simpler boosting algorithm. Gradient boosting is a flexible algorithm that can work with a variety of data types, including continuous, categorical, and binary data.

Gradient Boosting Regressor Algorithm works by forming an ensemble of weak prediction models, typically decision trees, and combining their predictions to form a more accurate model. In Gradient boosting regressors, the decision trees are built sequentially, with each subsequent tree fitting the residual errors of the preceding tree.

The Boosting algorithm places more emphasis on misclassified data points in subsequent models, which helps in minimizing prediction errors. Gradient Boosting Regressor Algorithm adds additional emphasis on the errors themselves, as compared to simple boosting. This is done by applying gradient descent optimization on loss functions to minimize the error.

The algorithm starts by fitting an initial model on the data. The residual errors of this model are then found, which are the differences between the predicted values and the actual values. A new model is built on these residual errors, and the two models are combined to improve the performance of the model.

In each round of the gradient boosting algorithm, a new model is built with the aim of minimizing the gradient of the loss function. The gradient is calculated using the derivative of the loss function with respect to the parameters of the model. The new model is then added to the ensemble of models, and the process is repeated until a predefined stopping criterion is reached.

The Gradient Boosting Regressor Algorithm has several advantages over other regression algorithms. Firstly, it has high prediction accuracy and low variance. Secondly, it can work with a variety of data types. Lastly, it can handle missing data and outliers.

In conclusion, Gradient Boosting Regressor Algorithm is one of the most powerful algorithms in the family of Boosting algorithms. It provides high accuracy and low variance, making it an ideal choice for many regression problems

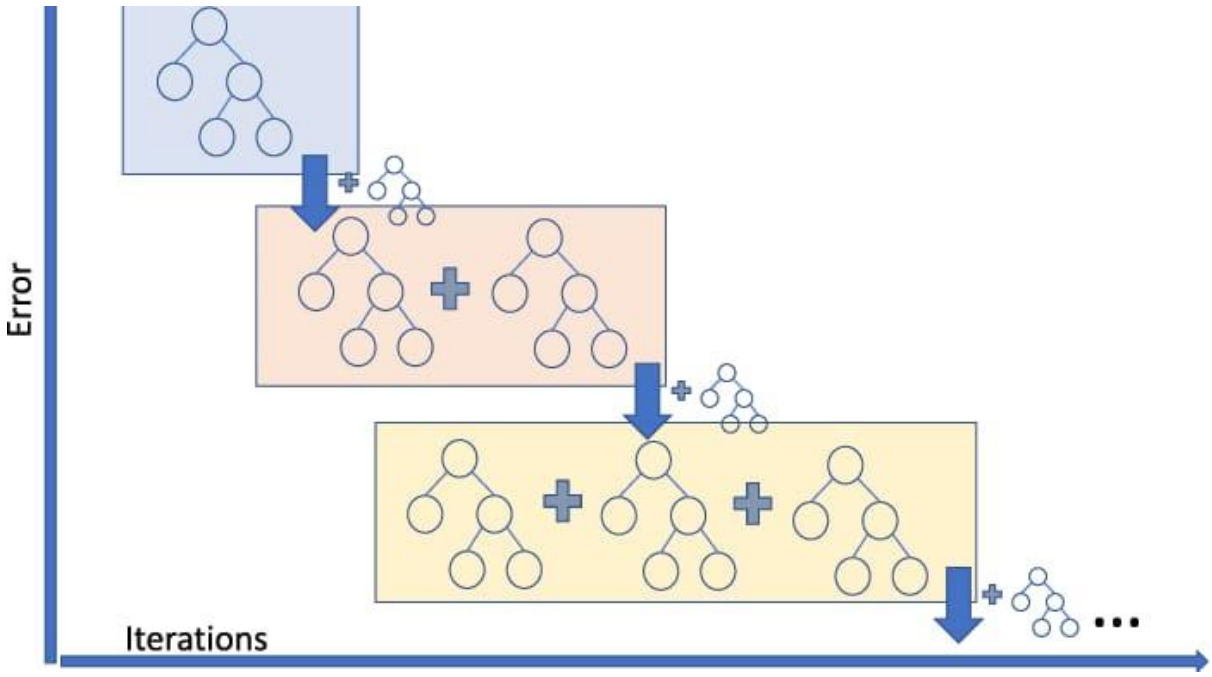


Fig 4 : Demonstration of Gradient Boosting Algorithm

CHAPTER 2

MERITS AND DEMERITS OF THE BASE PAPER

In the literature survey. A lot of works showing low performance, a few taking more time for training and some of them adequate performance metrics not used to evaluate the effectiveness of the method.

2.1 Literature review:

Following are some of the research papers which came up with various methods for Ayurveda herbs classification.

1. E.Agiree-basurko, G.IbarraBerastegi,I.Madoriaga had done research on the dataset of air pollution network dataset from environmental department basque government by using methods of regression and multilayer perception with limitations of each independent variables is affected by dependent variables ,computations are difficult and time consuming.

2. Thirupathi Boningani,Panagiotis G Smirniotis had done the research on the dataset of Customised dataset from American chemical soceity by proposing methods of Selective analyst reduction using classification in ML having limitations of There is no selection of NOx nitrogen in presence of excess of oxygen and Sulphur dioxide with broad temperature window which declines efficiency.

3. Muhammad Azjher hassan,Zhaomin Ding had done the research with dataset of Dataset from China National Environmental Monitoring Center by proposing methods of Artificial neutral network,MLP and GRNM having limitations of Large dataset would be profoundly efficient.

4. Huixiang Liu,Qing Li,Dongbing Yu,Yu Gu had done the research on dataset The Beijing air quality dataset from the Beijing environmental monitoring center by proosing the methods of SVR(Support Vector) RFR(Random Forest) having limitations of The SVR has increased time complexity and is not suitable for processing a large number of samples.

5. XiayongNi H.Huang N.P.Du had done the research on dataset biejing meteto logical data by proposing methods of multivariate statistical analysis method,BP neural network model,ARIMAtime series model having liimitations of it is a shorter term prediction study.long term prediction of pm(2.5) in the extent of months of years is not explored.

6. Shwet Ketu,Pramod Kumar Mishra had done the research on dataset of CPBC dataset of Delhi by proposing the methods of SVM classification algorithm with AKS(adjusting kernal scalling method) having limitations like computationally expensive &time consuming,problems with larger datasets.

2.2 Merits and Demerits of Proposed System:

Merits

1.As we have moved from decision tree, then random forest ,then gradient boosting regressor, this system has tried to improve the features which previous models have fallen prey for ,like multicollinearity and fitting problems.

2.low variance.

3.Accurate model.

4.This model trains faster even when the datasets ate larger.

5.Highly flexible and provides hyper parameter tuning.

Demerits

1.This model is not recommended when you don't have adequate compute power and training time.

2.Noisy dataset cannot be used as this system focusses on the tiniest error and it results in overfitting.

CHAPTER 3

SOURCE CODE

```
Import libraries and loading data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings(action='ignore')
from statsmodels.tsa.seasonal import seasonal_decompose
import statsmodels.formula.api as formula
from statsmodels.stats.outliers_influence import
variance_inflation_factor
import statsmodels as sm
from sklearn import preprocessing
from sklearn.model_selection import
train_test_split,GridSearchCV,RandomizedSearchCV
from sklearn.linear_model import LinearRegression,Lasso,Ridge
from sklearn.metrics import
r2_score,mean_squared_error,mean_absolute_error,median_absolute_error,mea
n_absolute_percentage_error,mean_squared_log_error,explained_variance_sco
re
from sklearn.ensemble import
RandomForestRegressor,GradientBoostingRegressor
from sklearn.tree import DecisionTreeRegressor
from IPython.display import display
import matplotlib.pyplot as plt



# get the name of the csv file
file ='https://raw.githubusercontent.com/prince381/air-
pollution/master/data/PRSA_Data_Wanshouxigong_20130301-20170228.csv'


# read the csv file into a pandas DataFrame using the pd.read_csv()
data = pd.read_csv(file)
data.head()

Data Preprocessing

# drop the unwanted columns/features
cols_to_drop = ['No','station']
data = data.drop(cols_to_drop,axis=1)
# print out the info of the data
data.info()



# check for duplicated values and null values
print('Are there any duplicated values in our data ? :
{}\n'.format(data.duplicated().any()))
print('The total number of null values in each colum:')
display(data.isnull().sum())

# find the most appearing wind direction value
data.wd.mode()
```

```
# fill in the missing values with the mean of the particular column
data.fillna(value=data.mean(),inplace=True)
# replace the missing values for the wind direction with the modal value
data.wd.fillna(value='NE',inplace=True)
# let's check the data again if there are any missing values
data.isnull().any()

# create a datetime column using the year,month,day and hour columns.
years = data['year'].values
months = data['month'].values
days = data['day'].values
hours = data['hour'].values
full_date = []

for i in range(data.shape[0]):
    date_time = str(years[i])+'-'+str(months[i])+'-'+str(days[i])+'
'+str(hours[i])+':'+str(0)
    full_date.append(date_time)

dates = pd.to_datetime(full_date)
dates = pd.DataFrame(dates,columns=['date'])
data = pd.concat([dates,data],axis=1)

data.head()

""" EXPLORATORY DATA ANALYSIS"""

plt.figure(figsize=(12,5))
sns.distplot(data['PM2.5'],bins=50)
plt.title('Distribution of the hourly recorded PM2.5 concetration in the
air\nin Aotizhongxin-Beijin',
          fontsize=16)
plt.show()

"""<h5><b>what pattern does the amount of PM2.5 concentration in the air
recorded in an hour follow
        for a daily time period ?</b></h5>
"""

# find the daily average of PM2.5 contained in the air in any given hour
daily_data = data[['date','PM2.5']]
daily_data = daily_data.set_index('date')
daily_data = daily_data.resample('D').median()
decomposition = seasonal_decompose(daily_data,model='addictive')

# plot the data
with plt.style.context('fivethirtyeight'):
    decomposition.trend.plot(figsize=(12,5),style='k-
',linewidth=.9,legend=False)
    plt.xlabel('Date',fontsize=14)
    plt.ylabel('PM2.5 concentration (ug/m^3)',fontsize=14)
    plt.title('Daily trend in the hourly recorded PM2.5 concentration
in\nthe air in Aotizhongxin-Beijin',fontsize=16)
    plt.grid(axis='x')
    plt.tight_layout()
    plt.show()

In which month does the amount of PM2.5 contained in the air rises ?
```

```
monthly_data = data[['month','PM2.5']]
months = ['January','February','March','April','May','June','July',
          'August','September','October','November','December']
ordered_monthdf = pd.DataFrame(months,columns=['month'])
map_dict = {}
for i,j in enumerate(months):
    map_dict.setdefault(i+1,j)

monthly_data.month = monthly_data.month.map(map_dict)
monthly_average = monthly_data.groupby('month').median()
monthly_average =
pd.merge(ordered_monthdf,monthly_average,left_on='month',right_index=True
)
monthly_average = np.round(monthly_average,1)
monthly_average = monthly_average.set_index('month')

# plot the data
with plt.style.context('ggplot'):

monthly_average.plot(figsize=(12,5),legend=False,kind='bar',linewidth=.9)
    plt.xlabel('Month',fontsize=14)
    plt.ylabel('PM2.5 concentration (ug/m^3)',fontsize=14)
    plt.title('Monthly average of the hourly recorded PM2.5 concentration
in\nthe air in Aotizhongxin-Beijin',fontsize=16)
    plt.grid(axis='x')
    plt.tight_layout()
    plt.show()

At what time of the day do we expect the amount of PM2.5 concentration in
the air to be high ?

hourly_data = data[['hour','PM2.5']]
hrs = ['12 AM','1 AM','2 AM','3 AM','4 AM','5 AM','6 AM','7 AM','8 AM','9
AM','10 AM',
      '11 AM','12 PM','1 PM','2 PM','3 PM','4 PM','5 PM','6 PM','7 PM',
      '8 PM','9 PM','10 PM','11 PM']
hour_dict = {}
for i,j in enumerate(hrs):
    hour_dict.setdefault(i,j)

hourly_data = hourly_data.groupby('hour').median().reset_index()
hourly_data.hour = hourly_data.hour.map(hour_dict)
hourly_data = hourly_data.set_index('hour')

# plot the data
with plt.style.context('ggplot'):

hourly_data.plot(figsize=(12,8),legend=False,kind='barh',linewidth=.9)
    plt.ylabel('Hours',fontsize=14)
    plt.xlabel('PM2.5 concentration (ug/m^3)',fontsize=14)
    plt.title('Average recorded PM2.5 concentration in the air in
Aotizhongxin-Beijin\nby the hour of the day',fontsize=16)
    plt.grid(axis='y')
    plt.tight_layout()
    plt.show()

In which direction does polluted air/wind mostly move ?
```

```
wind_dir = data[['wd','PM2.5']]
wind_dir = wind_dir.groupby('wd').median()

# plot the data
with plt.style.context('ggplot'):
    wind_dir.plot(figsize=(12,5),legend=False,kind='bar',linewidth=.9)
    plt.xlabel('Wind direction',fontsize=14)
    plt.ylabel('PM2.5 concentration (ug/m^3)',fontsize=14)
    plt.title('Average hourly recorded PM2.5 concentration in the air in
Aotizhongxin-Beijin\ngrouped by wind direction',fontsize=16)
    plt.grid(axis='x')
    plt.tight_layout()
    plt.show()

How do the other environmental factors affect the amount of PM2.5
concentration in the air ?

# let's try and visualize the relationships between the features of the
data
plt.figure(figsize=(13,9))
correlation_data = data[['PM2.5', 'PM10', 'SO2', 'NO2',
                         'CO', 'O3', 'TEMP', 'PRES',
                         'DEWP', 'RAIN', 'WSPM']]
sns.heatmap(correlation_data.corr(),cmap=plt.cm.Reds,annot=True)
plt.title('Heatmap displaying the correlation matrix of the
variables',fontsize=16)
plt.show()

Model Training and Evaluation
check for multicollinearity among variables and fit a regression model
using statsmodels
"""

cols_to_drop = ['date','year','month','day','hour','wd']
newdata = data.drop(cols_to_drop,axis=1)

# calculate the variance inflation factor of each feature and detect
multicollinearity
cons_data = sm.tools.add_constant(newdata)
series_before = pd.Series([variance_inflation_factor(cons_data.values,i)
for i in range(cons_data.shape[1])],
                          index=cons_data.columns)
series_before

# we can see that TEMP (temperature) and DEWP (dewpoint) are highly
correlated as the VIF value is
# greater than 5. As a result, we get rid of one of those features and
probably the one that has the
# lowest correlation with the dependent variable.
newdata = newdata.drop('DEWP',axis=1)
cons_data2 = sm.tools.add_constant(newdata)
series_after = pd.Series([variance_inflation_factor(cons_data2.values,i)
for i in range(cons_data2.shape[1])],
                          index=cons_data2.columns)
series_after
```

```
newdata.columns =
['PM2_5','PM10','SO2','NO2','CO','O3','TEMP','PRES','RAIN','WSPM']

# PM2.5 is skewed to the right so we log transform the values to
normalize the distribution
newdata['PM2_5'] = np.log(newdata['PM2_5'])

# fit the regression model
mul_reg = formula.ols(formula='PM2_5 ~ PM10 + SO2 + NO2 + CO + O3 + TEMP
+ PRES + RAIN + WSPM',
                      data=newdata).fit()
mul_reg.summary()

The OLS model from statsmodels gives us an accuracy of 71% (0.712) which
is not satisfactory for prediction. So we move on to fit a linear
regression model from the scikit-learn library.
fitting a linear regression model with
sklearn.linear_model.LinearRegression()
"""

# we split the data into predictor variables and Outcome variable
X = newdata.drop('PM2_5',axis=1)
y = newdata['PM2_5']

# we need to scale or normalize the predictor variables since they are
not on the same
# scale and some of their distributions are skewed.
X_scaled =  preprocessing.scale(X)
X_scaled = pd.DataFrame(X_scaled,columns=X.columns)
X_scaled.dropna(inplace=True)
# print the scaled predictor variables.
X_scaled.head()

# we now split out data into train and test data
X_train,X_test,y_train,y_test =
train_test_split(X_scaled,y,test_size=.2,random_state=0)

# instantiate the linear regression model
lin_model = LinearRegression()
lin_model.fit(X_train,y_train)   # fit the model

# we now score the model
print('Score on train data:
{}\n'.format(lin_model.score(X_train,y_train)))
print('Score on test data: {}'.format(lin_model.score(X_test,y_test)))

prediction = lin_model.predict(X_test)
mse = mean_squared_error(y_test,prediction)
accuracy = r2_score(y_test,prediction)
mae = mean_absolute_error(y_test, prediction)
medae = median_absolute_error(y_test, prediction)
mape = mean_absolute_percentage_error(y_test, prediction)
msle = mean_squared_log_error(y_test, prediction)
ev = explained_variance_score(y_test, prediction)
accuracy = r2_score(y_test,prediction)
```

```
print('Mean Squared Error: {}\n'.format(mse))
print('mean_absolute_error: {}'.format(mae))
print('median_absolute_error: {}'.format(medae))
print('mean_absolute_percentage_error: {}'.format(mape))
print('mean_squared_log_error: {}'.format( msle))
print('explained_variance_score: {}'.format(ev))
print('Overall model accuracy: {}'.format(accuracy))

The model accuracy for the LinearRegression() is no better than that of
the statsmodels. They all give the same accuracy is not better for making
predictions. We now move on to fit other models by using the ensemble
methods
Ensemble methods

"""

ensemble_data = data.drop(cols_to_drop,axis=1)

# we split the data into predictor variables and Outcome variable
X = ensemble_data.drop('PM2.5',axis=1)
y = ensemble_data['PM2.5']

xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=.2)
 """### DecisionTreeRegressor
we will now fit a decision tree regression model on the data and tune
some of its parameters to increase the accuracy."""

# we go ahead to use the ensemble methods as the LinearRegression model
has a low accuracy
# on both the test and train data.
decision_tree = DecisionTreeRegressor(max_depth=5,
                                      max_features='auto',
                                      min_samples_split=3,
                                      min_samples_leaf=2)
decision_tree.fit(xtrain,ytrain)

print('Score on train data:
{}\n'.format(decision_tree.score(xtrain,ytrain)))
print('Score on test data:
{}\n'.format(decision_tree.score(xtest,ytest)))

tree_pred = decision_tree.predict(xtest)


tree_ev = explained_variance_score(y_test, tree_pred)
tree_accuracy = r2_score(ytest,tree_pred)
tree_mse = mean_squared_error(ytest,tree_pred)
tree_mae = mean_absolute_error(y_test, tree_pred)
tree_medae = median_absolute_error(y_test, tree_pred)
tree_mape = mean_absolute_percentage_error(y_test, tree_pred)
tree_msle = mean_squared_log_error(y_test, tree_pred)



print('Root Mean Squared Error: {}\n'.format(np.sqrt(tree_mse)))
print('mean_absolute_error: {}'.format(tree_mae))
print('median_absolute_error: {}'.format(tree_medae))
```

```
print('mean_absolute_percentage_error: {}'.format(tree_mape))
print('mean_squared_log_error: {}'.format(tree_msle))
print('explained_variance_score: {}'.format(tree_ev))
print('Overall model accuracy: {}'.format(tree_accuracy))

# We now tune the parameters of the model to see if we can increase the
accuracy
params = {'max_depth':[3,4,5,6,7],
         'max_features':['auto','sqrt','log2'],
         'min_samples_split':[2,3,4,5,6,7,8,9,10],
         'min_samples_leaf':[2,3,4,5,6,7,8,9,10]}

tree = DecisionTreeRegressor()

# initialize the grid search for the best parameters
tree_search = GridSearchCV(tree,param_grid=params,
                           n_jobs=-1,cv=5)

tree_search.fit(xtrain,ytrain)   # fit the model

# we now score the model
print('Score on train data:
{}\n'.format(tree_search.score(xtrain,ytrain)))
print('Score on test data: {}\n'.format(tree_search.score(xtest,ytest)))
print('Best parameters found:')
display(tree_search.best_params_)

tree_search_pred = tree_search.predict(xtest)
tree_search_mse = mean_squared_error(ytest,tree_search_pred)
tree_search_mae = mean_absolute_error(ytest,tree_search_pred)
tree_search_medae = median_absolute_error(ytest,tree_search_pred)
tree_search_mape = mean_absolute_percentage_error(ytest,tree_search_pred)
tree_search_msle = mean_squared_log_error(ytest,tree_search_pred)
tree_search_ev = explained_variance_score(ytest,tree_search_pred)
tree_search_accuracy = r2_score(ytest,tree_search_pred)

print('Root Mean Squared Error: {}\n'.format(np.sqrt(tree_search_mse)))
print('mean_absolute_error: {}'.format(tree_search_mae))
print('median_absolute_error: {}'.format(tree_search_medae))
print('mean_absolute_percentage_error: {}'.format(tree_search_mape))
print('mean_squared_log_error: {}'.format(tree_search_msle))
print('explained_variance_score: {}'.format(tree_search_ev))
print('Overall model accuracy: {}'.format(tree_search_accuracy))

"""### RandomForestRegressor

we now fit a random forest regression model on the data to see if we
would get a better accuracy results than that of the decision tree
regression model.
"""

# instantiate the RandomForestRegressor model and fit the model on the
training data
forest = RandomForestRegressor(n_estimators=100,
                               max_depth=7,
                               max_features='auto',
                               min_samples_split=7,
                               min_samples_leaf=3)
```

```
forest.fit(xtrain,ytrain)

# we now score the model
print('Score on train data: {}\n'.format(forest.score(xtrain,ytrain)))
print('Score on test data: {}\n'.format(forest.score(xtest,ytest)))

forest_pred = forest.predict(xtest)
forest_mse = mean_squared_error(ytest,forest_pred)
forest_mae = mean_absolute_error(ytest,forest_pred)
forest_medae = median_absolute_error(ytest,forest_pred)
forest_mape = mean_absolute_percentage_error(ytest,forest_pred)
forest_msle = mean_squared_log_error(ytest,forest_pred)
forest_ev = explained_variance_score(ytest,forest_pred)
forest_accuracy = r2_score(ytest,forest_pred)

print('Root Mean Squared Error: {}\n'.format(np.sqrt(forest_mse)))
print('mean_absolute_error: {}'.format(forest_mae))
print('median_absolute_error: {}'.format(forest_medae))
print('mean_absolute_percentage_error: {}'.format( forest_mape))
print('mean_squared_log_error: {}'.format(forest_msle))
print('explained_variance_score: {}'.format(forest_ev))
print('Overall model accuracy: {}'.format(forest_accuracy))

# we now tune the parameters of the RandomForestRegressor model using
RandomizedSearchCV to
# find the best parameters and increase the accuracy of the model

params['n_estimators'] = [100,200,300,400,500]

# instantiate the model
random_forest = RandomForestRegressor()

# perform the grid search for the best parameters
forest_search = RandomizedSearchCV(random_forest,params,n_jobs=-1,
                                   cv=5,verbose=2)
forest_search.fit(xtrain,ytrain)

# we now score the model
print('Score on train data:
{}\n'.format(forest_search.score(xtrain,ytrain)))
print('Score on test data:
{}\n'.format(forest_search.score(xtest,ytest)))
print('Best parameters found:')
display(forest_search.best_params_)

forest_search_pred = forest_search.predict(xtest)
forest_search_mse = mean_squared_error(ytest,forest_search_pred)
forest_search_mae = mean_absolute_error(ytest,forest_search_pred)
forest_search_medae = median_absolute_error(ytest,forest_search_pred)
forest_search_mape =
mean_absolute_percentage_error(ytest,forest_search_pred)
forest_search_msle = mean_squared_log_error(ytest,forest_search_pred)
forest_search_ev = explained_variance_score(ytest,forest_search_pred)
forest_search_accuracy = r2_score(ytest,forest_search_pred)

print('Root Mean Squared Error: {}\n'.format(np.sqrt(forest_search_mse)))
print('mean_absolute_error: {}'.format(forest_search_mae))
```

```
print('median_absolute_error: {}'.format(forest_search_medae))
print('mean_absolute_percentage_error: {}'.format(forest_search_mape))
print('mean_squared_log_error: {}'.format(forest_search_msle))
print('explained_variance_score: {}'.format(forest_search_ev))
print('Overall model accuracy: {}'.format(forest_search_accuracy))

"""### GradientBoostingRegressor

we now fit a gradient boosting regression model on the data to see if we
would get a better accuracy results than that of the decision tree and
random forest regression model and also minimize the error.
"""

# instantiate the GradientBoostingRegressor model and fit the model on
the training data
grad_boost = GradientBoostingRegressor(n_estimators=100,
                                       max_depth=7,
                                       max_features='auto',
                                       min_samples_split=7,
                                       min_samples_leaf=3,
                                       learning_rate=0.1)

grad_boost.fit(xtrain,ytrain)

# we now score the model
print('Score on train data:
{}\n'.format(grad_boost.score(xtrain,ytrain)))
print('Score on test data: {}\n'.format(grad_boost.score(xtest,ytest)))

gboost_pred = grad_boost.predict(xtest)
gboost_mse = mean_squared_error(ytest,gboost_pred)
gboost_mae = mean_absolute_error(ytest,gboost_pred)
gboost_medae = median_absolute_error(ytest,gboost_pred)
gboost_mape = mean_absolute_percentage_error(ytest,gboost_pred)
gboost_ev = explained_variance_score(ytest,gboost_pred)
gboost_accuracy = r2_score(ytest,gboost_pred)

print('Root Mean Squared Error: {}\n'.format(np.sqrt(gboost_mse)))
print('mean_absolute_error: {}'.format(gboost_mae))
print('median_absolute_error: {}'.format(gboost_medae))
print('mean_absolute_percentage_error: {}'.format(gboost_mape))
print('explained_variance_score: {}'.format(gboost_ev))
print('Overall model accuracy: {}'.format(gboost_accuracy))

# we now tune the parameters of the GradientBoostingRegressor model using
RandomizedSearchCV to
# find the best parameters and increase the accuracy of the model
params['learning_rate'] = np.linspace(0.1,1,10)

# instantiate the model
gradient_boosting = GradientBoostingRegressor()

# perform the grid search for the best parameters
gboost_search = RandomizedSearchCV(gradient_boosting,params,n_jobs=-1,
                                   cv=5,verbose=2)
gboost_search.fit(xtrain,ytrain)

# we now score the model
```

```
print('Score on train data:
{}\n'.format(gboost_search.score(xtrain,ytrain)))
print('Score on test data:
{}\n'.format(gboost_search.score(xtest,ytest)))
print('Best parameters found:')
display(gboost_search.best_params_)

gboost_search_pred = gboost_search.predict(xtest)
gboost_search_mse = mean_squared_error(ytest,gboost_search_pred)
gboost_search_mae = mean_absolute_error(ytest,gboost_search_pred)
gboost_search_medae = median_absolute_error(ytest,gboost_search_pred)
gboost_search_mape =
mean_absolute_percentage_error(ytest,gboost_search_pred)
gboost_search_msle =
mean_squared_log_error(ytest,abs(gboost_search_pred))
gboost_search_ev = explained_variance_score(ytest,gboost_search_pred)
gboost_search_accuracy = r2_score(ytest,gboost_search_pred)

print('Root Mean Squared Error: {}\n'.format(np.sqrt(gboost_search_mse)))
print('mean_absolute_error: {}'.format(gboost_search_mae))
print('median_absolute_error: {}'.format(gboost_search_medae))
print('mean_absolute_percentage_error: {}'.format(gboost_search_mape))
print('mean_squared_log_error: {}'.format(gboost_search_msle))
print('explained_variance_score: {}'.format(gboost_search_ev))
print('Overall model accuracy: {}'.format(gboost_search_accuracy))

# we now use the best model (GradientBoostingRegressor model) to predict
the PM2.5
# concetration and compare it to the actual PM2.5 recorded in the data by
means of
# visualization

compare_data = pd.DataFrame({'dates':data['date'],
                             'Actual PM2.5':y,
                             'Predicted
PM2.5':gboost_search.predict(X.values)})

compare_data.set_index('dates',inplace=True)
compare_data['Predicted PM2.5'] = np.round(compare_data['Predicted
PM2.5'],1)

# let's plot the daily averages of the Actual PM10 and the predicted
PM2.5 concentration.
compare_data = compare_data.resample('D').mean()

with plt.style.context('fivethirtyeight'):
    plt.figure(figsize=(12,5))
    plt.scatter(compare_data.index,compare_data['Actual
PM2.5'],s=15,label='Actual PM2.5',
                alpha=.6)
    plt.scatter(compare_data.index,compare_data['Predicted
PM2.5'],s=15,label='Predicted PM2.5',
                alpha=.6)
    plt.legend()
    plt.title('Evaluating the GradientBoostingRegressor model\n(model
accuracy = 94%)',
              fontsize=18)
    plt.xlabel('period',fontsize=15)
```

```
    plt.ylabel('PM2.5 concentration',fontsize=15)
    plt.show()

Residuals analysis
Now that we have successfully trained a regression model that predicts
the amount of PM2.5 concetration in the air with a 93% accuracy given
other environmental features, we have to analyze the errors of prediction
to see if the model satisfies the regression errors asumption. That
is,the errors must be normally distributed and do not follow any pattern.


# calculate the errors
compare_data['Residuals'] = compare_data['Actual PM2.5'] -
compare_data['Predicted PM2.5']

# make a scatter plot of the errors to see if they follow any pattern
with plt.style.context('ggplot'):
    plt.figure(figsize=(12,5))
    plt.scatter(compare_data.index,compare_data.Residuals,alpha=.7)
    plt.title('Residual scatter plot',fontsize=16)
    plt.ylabel('Errors',fontsize=15)
    plt.grid(axis='x')
    plt.show()

# plot the histogram to see check the normality of the errors
plt.figure(figsize=(12,5))
sns.distplot(compare_data.Residuals,bins=50)
plt.title('Distribution of the residuals from the gradient boosting
model',
          fontsize=16)
plt.show()

As seen above, the residuals of the predictions follow no pattern and
also have a normal distribution which satisfies the regression error
assumptions. This proves that our model is accurate and good for further
predictions.

"""

rmse_scores = {'RandomForest': np.sqrt(forest_mse), ' DecisionTree':
np.sqrt(tree_mse), ' GradientBoosting': np.sqrt(gboost_mse)}


plt.bar(range(len(rmse_scores)), list(rmse_scores.values()),
align='center')
plt.xticks(range(len(rmse_scores)), list(rmse_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('RMSE')
plt.title('Comparison of RMSE scores for different algorithms')
plt.show()
print('Root Mean Squared Error in RandomForest:
{}\n'.format(np.sqrt(forest_mse)))
print('Root Mean Squared Error in DecisionTree:
{}\n'.format(np.sqrt(tree_mse)))
print('Root Mean Squared Error in GradientBoosting:
{}\n'.format(np.sqrt(gboost_mse)))
```

```
mae_scores = {'RandomForest':forest_mae , ' DecisionTree': tree_mae, '
GradientBoosting': gboost_mae}


plt.bar(range(len(mae_scores)), list(mae_scores.values()),
align='center')
plt.xticks(range(len(mae_scores)), list(mae_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('MeanAbsoluteError')
plt.title('Comparison of MAE scores for different algorithms')
plt.show()
print('mean_absolute_error in RandomForest: {}'.format(forest_mae))
print('mean_absolute_error in DecisionTree: {}'.format(tree_mae))
print('mean_absolute_error in GradientBoosting: {}'.format(gboost_mae))

medae_scores = {'RandomForest':forest_medae , ' DecisionTree':
tree_medae, ' GradientBoosting': gboost_medae}


plt.bar(range(len(medae_scores)), list(medae_scores.values()),
align='center')
plt.xticks(range(len(medae_scores)), list(medae_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('MedianAbsoluteError')
plt.title('Comparison of MEDAE scores for different algorithms')
plt.show()
print('median_absolute_error in RandomForest: {}'.format(forest_medae))
print('median_absolute_error in DecisionTree: {}'.format(tree_medae))
print('median_absolute_error in GradientBoosting:
{}'.format(gboost_medae))

mape_scores = {'RandomForest':forest_mape , ' DecisionTree': tree_mape, '
GradientBoosting': gboost_mape}


plt.bar(range(len(mape_scores)), list(mape_scores.values()),
align='center')
plt.xticks(range(len(mape_scores)), list(mape_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('mean_absolute_percentage_error')
plt.title('Comparison of MAPE scores for different algorithms')
plt.show()
print('mean_absolute_percentage_error in RandomForest:
{}'.format(forest_mape))
print('mean_absolute_percentage_error in DecisionTree:
{}'.format(tree_mape))
print('mean_absolute_percentage_error in GradientBoosting:
{}'.format(gboost_mape))

msle_scores = {'RandomForest':forest_msle , ' DecisionTree': tree_msle, '
GradientBoosting': gboost_msle}


plt.bar(range(len(msle_scores)), list(msle_scores.values()),
align='center')
plt.xticks(range(len(msle_scores)), list(msle_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel(' mean_squared_log_error')
```

```
plt.title('Comparison of MSLE scores for different algorithms')
plt.show()
print(' mean_squared_log_error in RandomForest: {}'.format(forest_msle))
print(' mean_squared_log_error in DecisionTree: {}'.format(tree_msle))
print(' mean_squared_log_error in GradientBoosting:
{}'.format(gboost_msle))

ev_scores = {'RandomForest':forest_ev , ' DecisionTree': tree_ev, '
GradientBoosting': gboost_ev}


plt.bar(range(len(ev_scores)), list(ev_scores.values()), align='center')
plt.xticks(range(len(ev_scores)), list(ev_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('explained_variance')
plt.title('Comparison of explained_variance scores for different
algorithms')
plt.show()
print(' mean_squared_log_error in RandomForest: {}'.format(forest_ev))
print(' mean_squared_log_error in DecisionTree: {}'.format(tree_ev))
print(' mean_squared_log_error in GradientBoosting:
{}'.format(gboost_ev))

accuracy_scores = {'RandomForest':forest_accuracy , ' DecisionTree':
tree_accuracy, ' GradientBoosting': gboost_accuracy}


plt.bar(range(len(accuracy_scores)), list(accuracy_scores.values()),
align='center')
plt.xticks(range(len(accuracy_scores)), list(accuracy_scores.keys()))
plt.xlabel('Algorithm')
plt.ylabel('overall accuracy')
plt.title('Comparison of overall accuracy scores for different
algorithms')
plt.show()
print(' mean_squared_log_error in RandomForest:
{}'.format(forest_accuracy))
print(' mean_squared_log_error in DecisionTree:
{}'.format(tree_accuracy))
print(' mean_squared_log_error in GradientBoosting:
{}'.format(gboost_accuracy))
```

CHAPTER 4

SNAP SHOTS

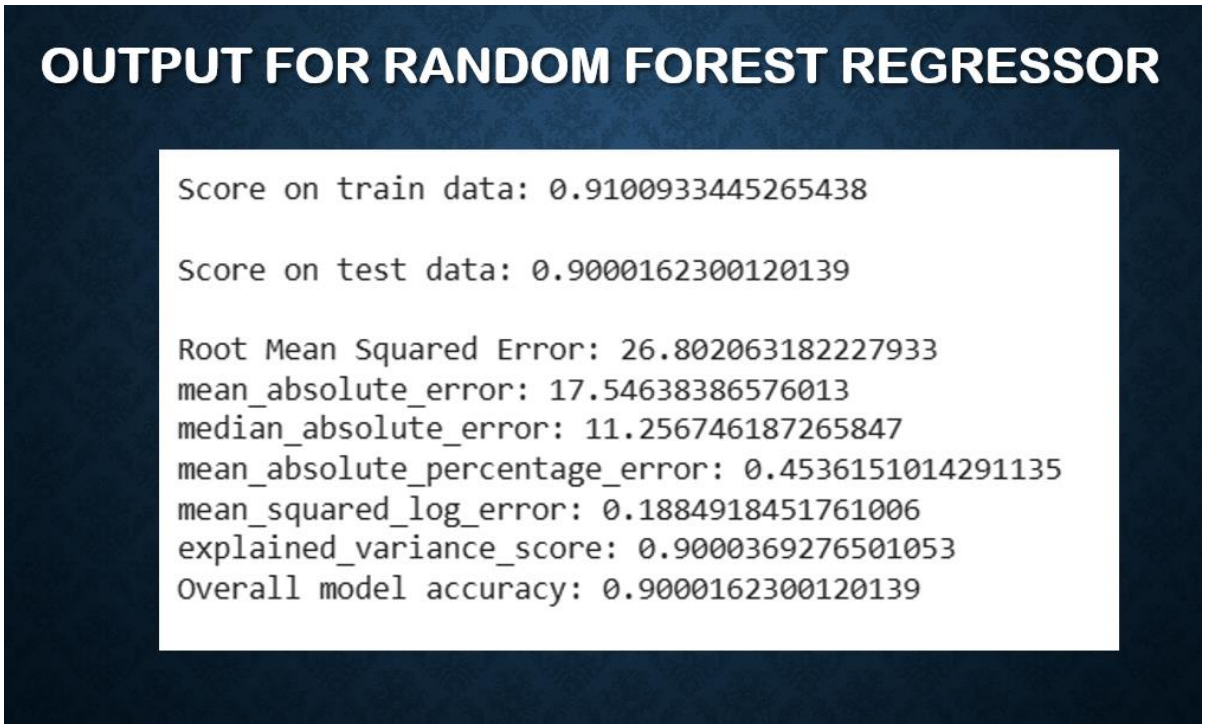


Fig 5

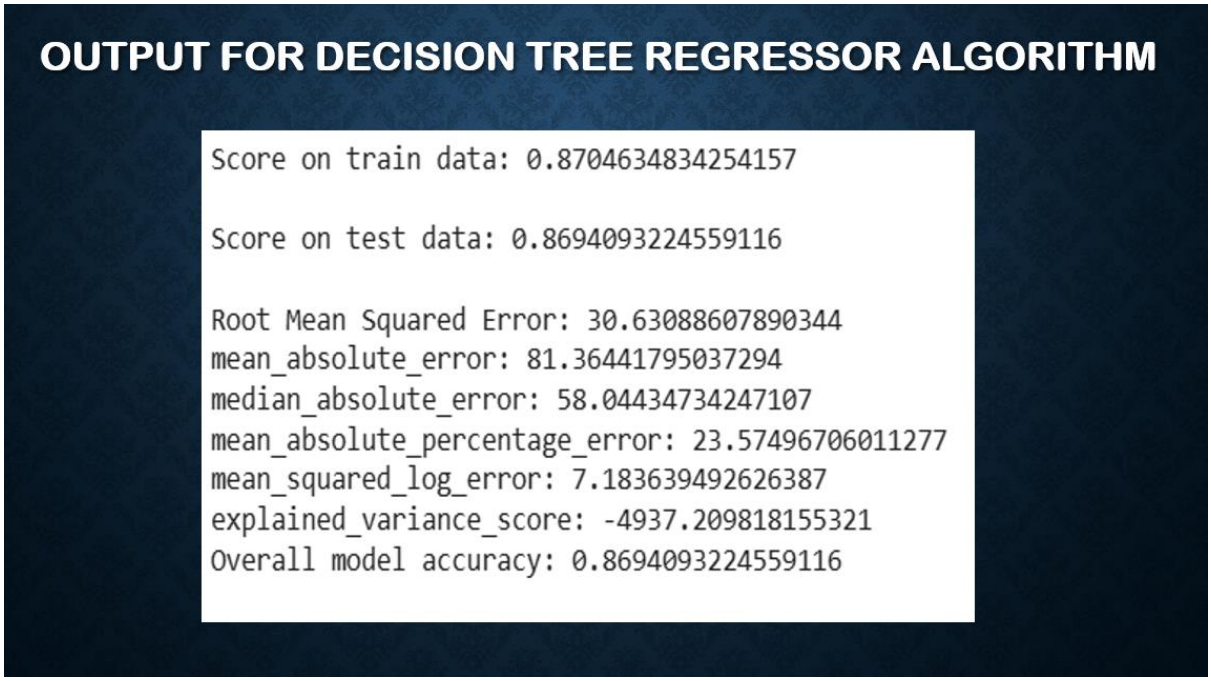


Fig 6

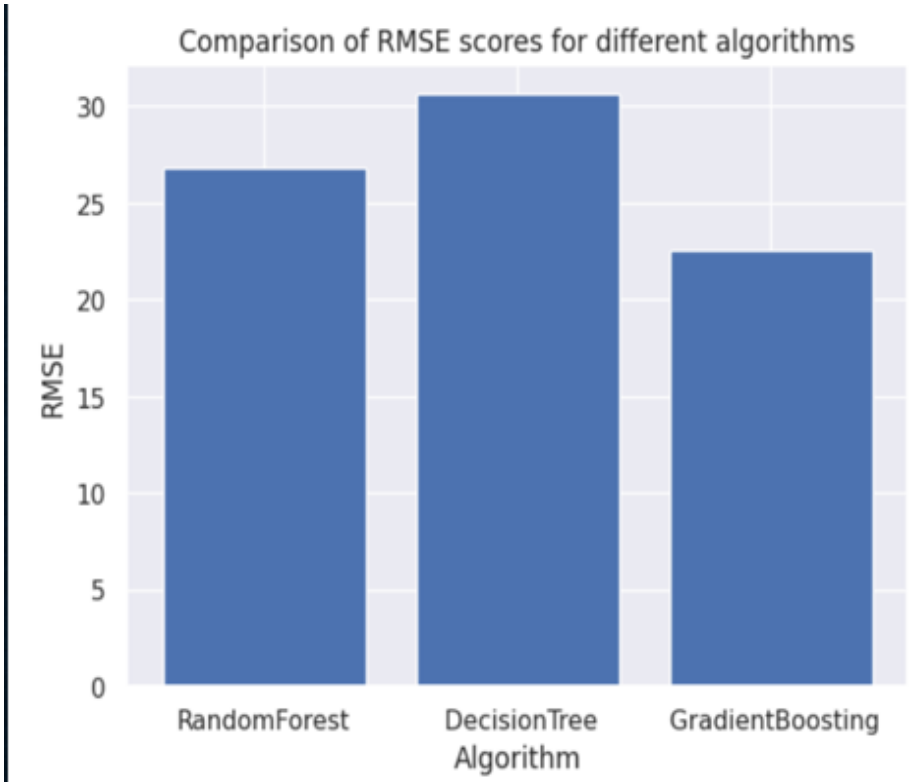
OUTPUT FOR GRADIENT BOOSTING REGRESSOR

```
Score on train data: 0.9609221137271996

Score on test data: 0.9294177028978993

Root Mean Squared Error: 22.519117486482298
mean_absolute_error: 14.51582783135057
median_absolute_error: 9.040192805850339
mean_absolute_percentage_error: 0.37067215578850976
mean_squared_log_error: 0.14848313538828792
explained_variance_score: 0.9294399232185303
Overall model accuracy: 0.9294177028978993
```

Fig 7




```
Root Mean Squared Error in RandomForest: 26.802063182227933

Root Mean Squared Error in DecisionTree: 30.63088607890344

Root Mean Squared Error in GradientBoosting: 22.519117486482298
```

Fig 8

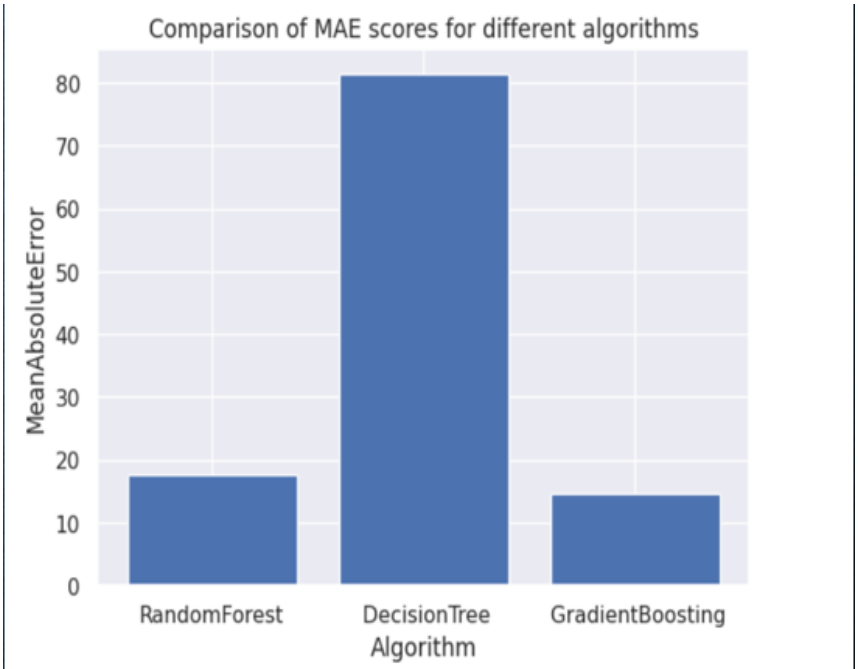

Comparison of MAE scores for different algorithms
MeanAbsoluteError
0
10
20
30
40
50
60
70
80
RandomForest
DecisionTree
GradientBoosting
Algorithm
mean_absolute_error in RandomForest: 17.54638386576013
mean_absolute_error in DecisionTree: 81.36441795037294
mean_absolute_error in GradientBoosting: 14.51582783135057

Fig 9

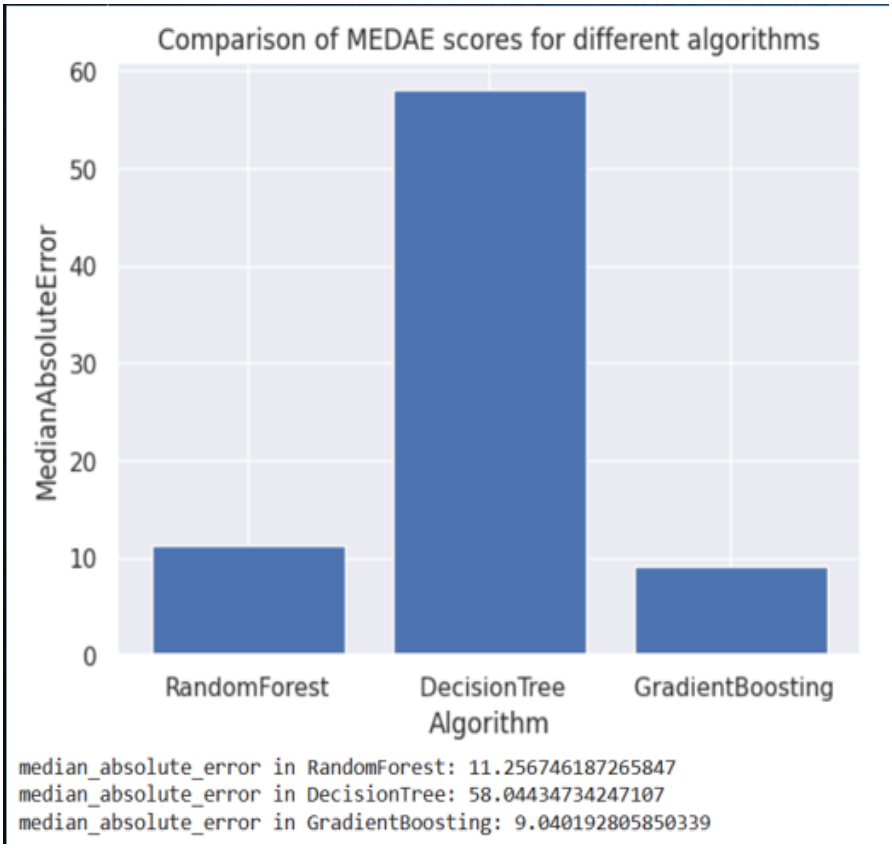
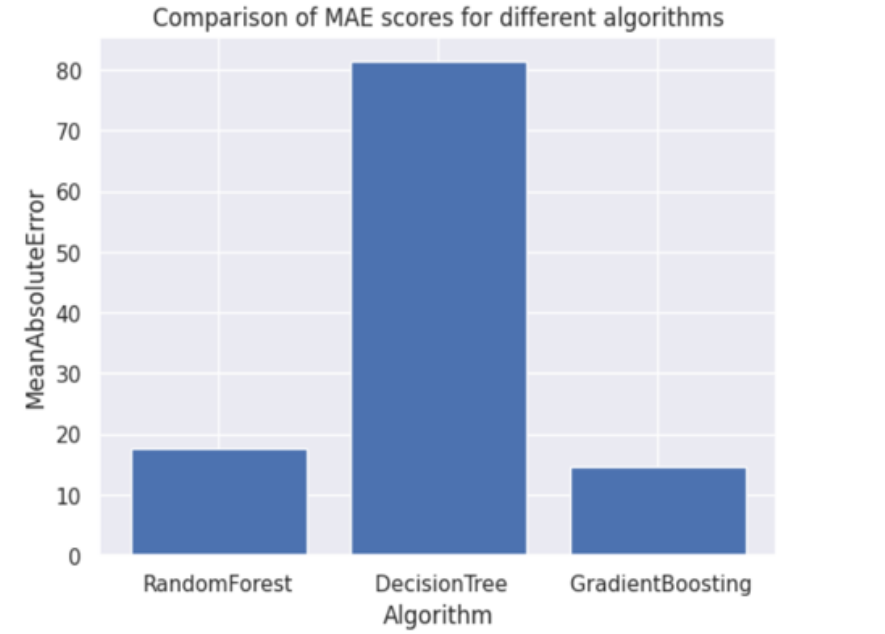
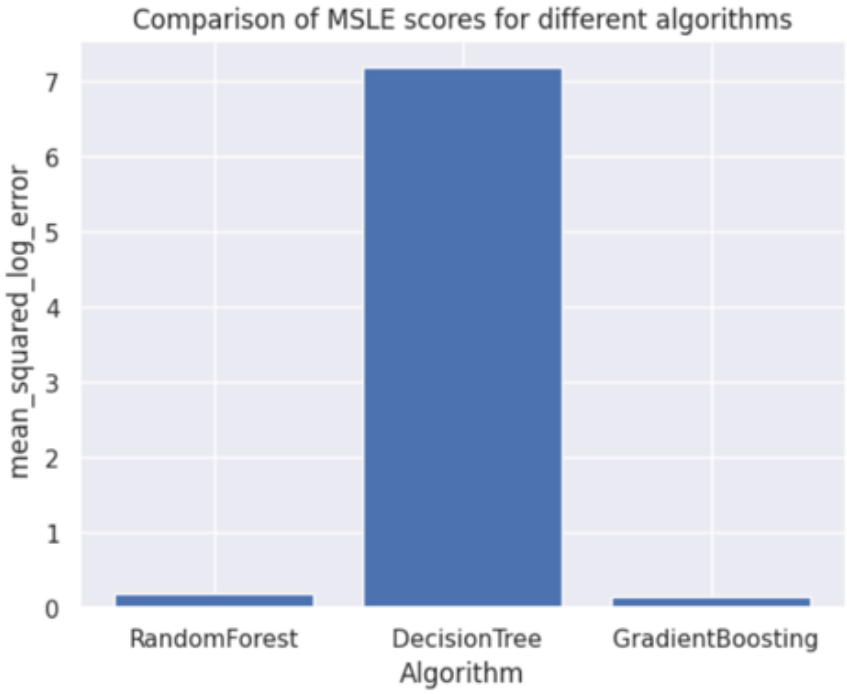

Comparison of MEDAE scores for different algorithms
MedianAbsoluteError
0
10
20
30
40
50
60
RandomForest
DecisionTree
GradientBoosting
Algorithm
median_absolute_error in RandomForest: 11.256746187265847
median_absolute_error in DecisionTree: 58.04434734247107
median_absolute_error in GradientBoosting: 9.040192805850339

Fig 10



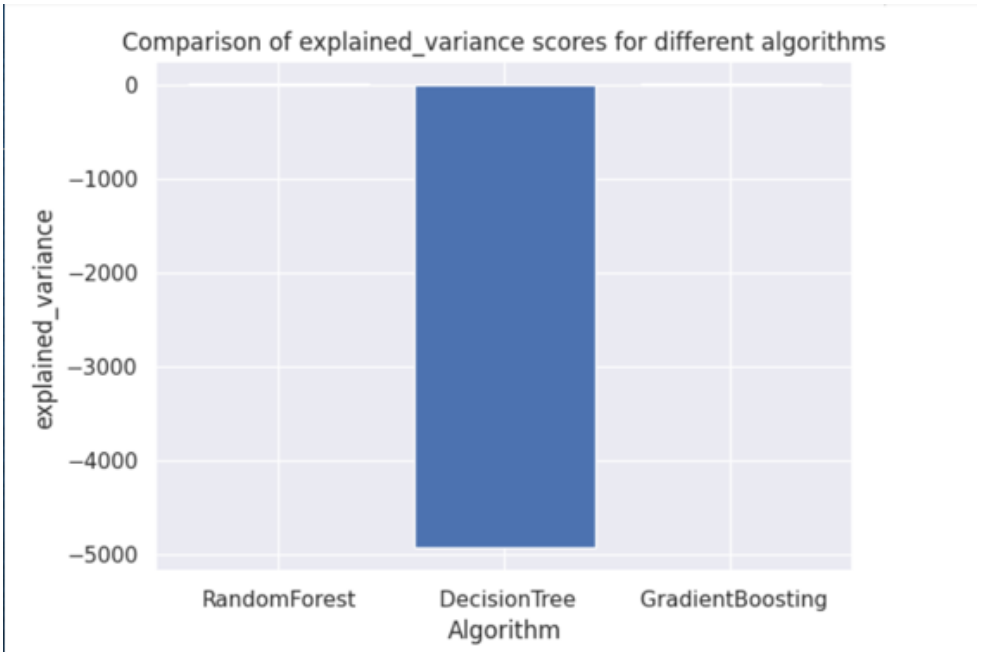

```
mean_absolute_error in RandomForest: 17.54638386576013
mean_absolute_error in DecisionTree: 81.36441795037294
mean_absolute_error in GradientBoosting: 14.51582783135057
```

Fig 11



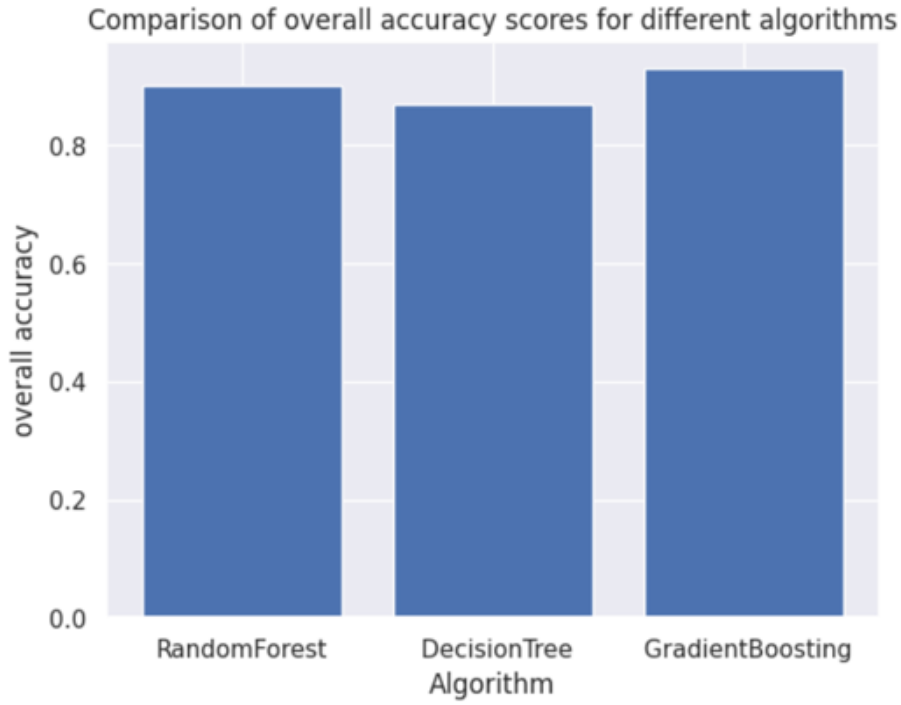

```
mean_squared_log_error in RandomForest: 0.1884918451761006
mean_squared_log_error in DecisionTree: 7.183639492626387
mean_squared_log_error in GradientBoosting: 0.14848313538828792
```

Fig 12




```
mean_squared_log_error in RandomForest: 0.9000369276501053
mean_squared_log_error in DecisionTree: -4937.209818155321
mean_squared_log_error in GradientBoosting: 0.9294399232185303
```

Fig 13




```
mean_squared_log_error in RandomForest: 0.9000162300120139
mean_squared_log_error in DecisionTree: 0.8694093224559116
mean_squared_log_error in GradientBoosting: 0.9294177028978993
```

Fig 14

CHAPTER 5

CONCLUSION AND FUTURE PLANS

5.1 Conclusion

Here, we used a machine learning-based model to predict airborne PM2.5 concentrations. We specifically compared Recursive Feature Screening with the Random Forest Regression (RFERF) model, the Decision Tree Regression model, and the Gradient Boosting Regression model

In order to increase public awareness of air quality and reduce the overall impact of adverse weather conditions on human health, this study highlights the importance of air quality forecasting.

Overall, this study offered promising ideas for machine learning-based weather forecasting in Indi

5.2 Future plan

Future work will emphasis on improving the model to give good performance even on the datsets which have more noise.It can also be improved to take less time for training by making fusion of algorithms but keeping the accuracy and flexibility intact by new hybridisation of algorithms.It can be made use with different type of datasets and their features and extend its predictions to different domains and sectors to greater human cause

CHAPTER 6

REFERENCES

1. Agirre-Basurko E, Ibarra-Berastegi G, Madariaga I (2006) Regression and multilayer perceptronbased models to forecast hourly O3 and NO2 levels in the Bilbao area. Environ Model Softw 21(4):430–446
2. Anderson JO, Thundiyil JG, Stolbach A (2012) Clearing the air: a review of the efects of particulate matter air pollution on human health. J Med Toxicol 8(2):166–175
3. Boningari T, Smirniotis PG (2016) Impact of nitrogen oxides on the environment and human health: Mnbased materials for the NOx abatement. Curr Opin Chem Eng 13:133–141
4. Cleland JG, Van Ginneken JK (1988) Maternal education and child survival in developing countries: the search for pathways of infuence. Soc Sci Med 27(12):1357–1368
5. Du X, Kong Q, Ge W, Zhang S, Fu L (2010) Characterization of personal exposure concentration of fine particles for adults and children exposed to high ambient concentrations in Beijing, China. J Environ Sci 22(11):1757–1764
6. India at a Glance (2019) Population enumeration data. https://www.india.gov.in/india-glance/profle.Accessed 9 Dec 2019
7. Ketu S, Mishra PK (2021c) Cloud, fog and mist computing in IoT: an indication of emerging opportunities IETE Tech Rev. https://doi.org/10.1080/02564602.2021.1898482
8. Ketu S, Mishra PK (2022d) A contemporary survey on IoT based smart cities: architecture, applications, and open issues. Wirel Person Commun. https://doi.org/10.1007/s11277-022-09658-2
9. Packtpub(2018) Machine learning algorithms. https://www.packtpub.com/product/machine-learningalgorithms-second edition/9781789347999. Accessed 15 May 2022
10. The World Bank (2019) Population total—India. https://data.worldbank.org/indicator/SP.POP.TOTL?locations=IN. Accessed 9 Dec 2019
11. Yan K, Zhang D (2015) Feature selection and analysis on correlated gas sensor data with recursive feature elimination. Sensors Actuators B Chem 212:353–363
12. Zhang Q, Jiang X, Tong D, Davis SJ, Zhao H, Geng G et al (2017) Transboundary health impacts of transported global air pollution and international trade. Nature 543(7647):705–709

CHAPTER 7

Appendix-Base paper

Natural Hazards (2022) 114:2109–2138
https://doi.org/10.1007/s11069-022-05463-z

ORIGINAL PAPER

Spatial Air Quality Index and Air Pollutant Concentration prediction using Linear Regression based Recursive Feature Elimination with Random Forest Regression (RFERF): a case study in India

Shwet Ketu[1]

Received: 25 January 2022 / Accepted: 20 June 2022 / Published online: 21 July 2022

Abstract
In the last decade, air pollution has become one of the vital environmental issues and has expanded its wings day by day. Prediction of air quality plays a crucial role in warning people about the air quality levels. With the help of this, we can make the proper mechanism for reducing the overall impact of bad air quality on individuals' health. In this paper, we are focused on developing a mechanistic and quantitative prediction model for the prediction of the Air Quality Index (AQI) and Air Pollutant Concentration (NOx) levels with a clear environmental interpretation. The proposed model is based on the Linear Regression based Recursive Feature Elimination with Random Forest Regression (RFERF). For the experimental analysis, the seven well-established machine learning models have been taken, and these models are compared with our proposed model to find out their suitability and correctness. The Mean Absolute Percentage Error (MAPE), mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and Coefficient of Determination ($R2$ score) have been used to validate the performance of prediction models. For the prediction of AQI and NOx, the data of the Central Pollution Control Board of India has been taken. The proposed model performs superior as compared to other prediction models with better accuracy and a higher prediction rate. This work also explains that linking machine learning with sensor-generated AQI data for air quality prediction is an adequate and appropriate way to solve some related environment glitches. Apart from this, the impact of air pollution on individuals' health due to high levels of AQI, NOx, and other pollutants with the possible solutions has also been covered.

Keywords Air Quality Index (AQI) · Machine learning · Air quality · Air pollutant concentration (NOx) · Prediction model

✉ Shwet Ketu
shwetiiita@gmail.com

[1] Department of Systemics, School of Computer Science, University Of Petroleum And Energy Studies(UPES), Dehradun, India

1 Introduction

In the current era, we are facing many environmental crises like hazardous waste, global warming, water pollution, air pollution, resource depletion, and much more (Vitousek 1994; Yilmaz et al. 2017; De Vito et al. 2009; Northey et al. 2018). Every year millions of people die due to the various diseases caused by air pollution (Zhang et al. 2017). In a developing country like India, the impact of air pollution on individuals' health is becoming a serious public health concern (Du et al. 2010). India is the second most highly populated country worldwide after China. It is having a population of around 1.34 Billion, which is a huge number (The World Bank 2019). Suppose we talk about Delhi, India's capital, and an industrial hub of the country. Thus, in the last couple of years, we have seen tremendous growth in the population of Delhi (India at a Glance 2019). A more population leads to more energy and water consumption as well as also plays a vital role in enhancing the pollution levels. Alongside this rapid development, the foggy weather caused by air pollution has become a serious matter of concern for Delhi. This foggy weather can be occurred often and may persist for a long time. Resultants, a rapid downfall has been seen in the air quality of Delhi in the last couple of years (Mishra 2019).

For evaluating the air quality, the Air Quality Index (AQI) is an important parameter used by government bureaus that tell the public how polluted the air is currently (Kyrkilis et al. 2007). According to the Indian National Ambient Air Quality Standards Standard (Chelani et al. 2002), AQI is calculated by six significant pollutants such as inhalable particles (PM10), fine particulate matter (PM2.5), nitrogen monoxide (NO), nitrogen oxide (NOx), nitrogen dioxide (NO_2), ammonia (NH_3), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3) (Fan et al. 1999). AQI measures the quality of the air in the range of 0 to 500, which is divided into six levels such as good, satisfactory, moderate, poor, very poor, and severely polluted (Deswal and Verma 2016). These levels illustrate the impact of various levels on the individuals' health and also give a worthy reference in a numerical format for the individuals' outdoor activities (Zhu et al. 2017). A lower level of AQI indicates good air quality, while a higher level indicates lousy air quality. The bad air quality may cause complications for the individuals' outdoor deeds.

Air Pollutant Concentration (NOx) are the toxic gases that are acquired from the combustion of oxygen and nitrogen under immense temperature and pressure. It is emitted by various resources such as the smoke of vehicles, industries, and household things. A high level of NOx is playing a vital role not only in air pollution but also affects human health directly and indirectly. Its direct impact on human health are headaches, breathing problems, eye irritation, chronically reduced lung function, corroded teeth, and loss of appetite. Indirectly, it can also harm the ecosystem by polluting water and land. It also contributes to ozone destruction, toxic fogs, suffocating smoke, and acid rain.

Good quality of air is a necessity for everyone. Thus, proper monitoring and handling of the problem of air pollution is an essential task. With the help of accurate air quality prediction, we can take preventive action accordingly and minimize the overall impact of air pollution on individuals. So, there is a need for efficient forecasting methods that can able to predict air quality. It will help the government sector as well as the private sector to take protective measures for preventing severe pollution incidents. With the help of forecasting results, various policies have been made by the government of Delhi to reduce air pollution levels. The Delhi government has adopted preventive actions such as banning the burning of leaves/ biomass, an odd–even policy for the 4-wheeler, and the construction of various expressways to reduce traffic overload (Ganguly et al. 2019).

Machine learning (ML) is based on the scientific study of statistical models and algorithms. Computer systems can use it for making decisions or predictions based on patterns and interfaces in the absence of any explicit instructions (Bishop 2006). Tremendous growth has been seen in the field of machine learning, and subsequently, it becomes a giant player in the field of data science. With the help of this, fast and effective prediction is possible for complex problems like big data (Ketu et al. 2015, 2020a; Ketu and Mishra 2015) and sensor-based data (stream data) (Ketu and Mishra 2020b, 2022a, c). Big data means an enormous amount of data. It consists of 5V's, which are velocity, variety, veracity, volume, and value (Packtpub 2018). Sensor-based data or stream data has become more popular in the field of data science and may consist of massive unstructured data that need real-time analysis (Ketu and Mishra 2021a, b, c). The solution to this type of problem can be possible only with machine learning algorithms (Chen et al. 2014).

Motivated by earlier studies, we found the two directions. In the first direction, we have explored the correlation among the various air indicators such as inhalable particles (PM10), fine particulate matter (PM2.5), nitrogen monoxide (NO), nitrogen oxide (NOx), nitrogen dioxide (NO_2), ammonia (NH_3), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3). In the second direction, we have compared our proposed RFERF model to the seven well-known machine learning models, i.e., Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), K-Nearest Neighbors Regression (KNNR), linear regression (LR), Multi-layer Perceptron Regression (MLPR), Random Forest Regression (RFR), and support vector regression (SVR). This rigorous comparison has been performed to find out the suitability and correctness of our proposed model. Finally, the performance evaluation of the regression models has been done by using various performance measures, i.e., Mean Absolute Percentage Error (MAPE), mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and Coefficient of Determination ($R2$ score).

The primary aim of this research work is:

- To develop a mechanistic and quantitative prediction model that will be suited for the forecasting of AQI and Air Pollutant Concentration (NOx) levels with a clear environmental interpretation.
- To find out the suitability and correctness of our proposed among the various well-established machine learning models.
- Apart from this, the impact of air pollution on individuals' health due to high levels of AQI, NOx, and other pollutants, along with the possible solutions.

The rest of this paper is organized as follows. In Sect. 2, the related work has been impersonated. The datasets and prediction models have been briefly described in Sect. 3. Section 4 shows the results based on the existing and proposed prediction models. A detailed discussion of the result has been presented in Sect. 5. Concluding remarks are drawn in Sect. 6.

2 Related Work

From time to time, various researchers had been successfully applied several machine learning models for long and short-term prediction of air quality (Cabaneros et al. 2019, 2017; Lightstone et al. 2017; Ibarra-Berastegi et al. 2008). Some investigators had

Machine learning (ML) is based on the scientific study of statistical models and algorithms. Computer systems can use it for making decisions or predictions based on patterns and interfaces in the absence of any explicit instructions (Bishop 2006). Tremendous growth has been seen in the field of machine learning, and subsequently, it becomes a giant player in the field of data science. With the help of this, fast and effective prediction is possible for complex problems like big data (Ketu et al. 2015, 2020a; Ketu and Mishra 2015) and sensor-based data (stream data) (Ketu and Mishra 2020b, 2022a, c). Big data means an enormous amount of data. It consists of 5V's, which are velocity, variety, veracity, volume, and value (Packtpub 2018). Sensor-based data or stream data has become more popular in the field of data science and may consist of massive unstructured data that need real-time analysis (Ketu and Mishra 2021a, b, c). The solution to this type of problem can be possible only with machine learning algorithms (Chen et al. 2014).

Motivated by earlier studies, we found the two directions. In the first direction, we have explored the correlation among the various air indicators such as inhalable particles (PM10), fine particulate matter (PM2.5), nitrogen monoxide (NO), nitrogen oxide (NOx), nitrogen dioxide (NO_2), ammonia (NH_3), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3). In the second direction, we have compared our proposed RFERF model to the seven well-known machine learning models, i.e., Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), K-Nearest Neighbors Regression (KNNR), linear regression (LR), Multi-layer Perceptron Regression (MLPR), Random Forest Regression (RFR), and support vector regression (SVR). This rigorous comparison has been performed to find out the suitability and correctness of our proposed model. Finally, the performance evaluation of the regression models has been done by using various performance measures, i.e., Mean Absolute Percentage Error (MAPE), mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and Coefficient of Determination ($R2$ score).

The primary aim of this research work is:

- To develop a mechanistic and quantitative prediction model that will be suited for the forecasting of AQI and Air Pollutant Concentration (NOx) levels with a clear environmental interpretation.
- To find out the suitability and correctness of our proposed among the various well-established machine learning models.
- Apart from this, the impact of air pollution on individuals' health due to high levels of AQI, NOx, and other pollutants, along with the possible solutions.

The rest of this paper is organized as follows. In Sect. 2, the related work has been impersonated. The datasets and prediction models have been briefly described in Sect. 3. Section 4 shows the results based on the existing and proposed prediction models. A detailed discussion of the result has been presented in Sect. 5. Concluding remarks are drawn in Sect. 6.

2 Related Work

From time to time, various researchers had been successfully applied several machine learning models for long and short-term prediction of air quality (Cabaneros et al. 2019, 2017; Lightstone et al. 2017; Ibarra-Berastegi et al. 2008). Some investigators had

proposed to predict objects into the discretized levels too. Pérez et al. used a multi-layer neural network for predicting the PM2.5 concentration recorded on an hourly basis in Santiago city (Pérez et al. 2000). For better prediction results, the authors suggested using a vast dataset. Two-hybrid models had been proposed by Zhu et al., which were EMD–intrinsic mode functions (IMF) based hybrid model and empirical mode decomposition (EMD)–SVR based hybrid model. These models were used to forecast the AQI of the Xingtai city, and with an accuracy of 80%, the EMD–SVR hybrid model had won the race (Zhu et al. 2017). This experiment had been performed to predict the current AQI level with the help of past AQI data using a single indicator only. The correlation among the various pollutants (such as PM10, PM2.5, and so on) had been ignored in the experiment. The FFNN (Feed-Forward Neural Networks) had been used by Corani et al. for the prediction of Ozone and PM10 in Milan city (Corani 2005). The prediction based on the FFNN model was satisfactory, and as per the author's report, the model had trended toward overfitting. The RNN (recursive neural network) model was used by Biancofiore et al. to forecast the concentration of PM10 (Biancofiore et al. 2017). The RNN model prediction was quite accurate, with a prediction accuracy of 95%. The false-positive percentage was 30%, which had shown the limits of the neural network model while simulating the concentration peaks. The empirical model was formulated by Fuller et al. for the prediction of PM10 concentrations in London city (Fuller et al. 2002). The data were collected from roadside and offside (background) locations in London. However, the overall performance of the empirical model was heavily dependent on the ratio between NOx and PM10.

To find out the effects of air pollution, Kalapanidas and Avouris (2001) had taken the climatological features such as precipitation, wind, temperature, humidity, and solar radiation. The classification of air pollution was done using CBR (case-based reasoning system) and the lazy learning approach to the various levels (low, mid, high, and alarm). For the prediction and categorization of the concentrations of O_3 in the environment, the s-fuzzy lattice neurocomputing classifier was used by Athanasiadis et al. (2003), and the concentrations of O_3 were categorized into three levels (low, medium, and high). The climatological features and other pollutants such as SO_2, NO_2, NO, and so on were taken to build the prediction model. Kurt and Oktay (2010) had established a geographic network into a neural network model for the prediction of CO, SO_2, and PM10 concentration levels in advance. Corani et al. evaluated the performance-based comparison among the pruned neural networks (PNNs) and feed-forward neural networks (FFNNs) for predicting the concentrations of O_3 and PM10 with the help of the previous day's data. More efforts on FFNNs had been applied by Fu et al. (Fu et al. 2015) for improving the existing FFNN models with the help of the gray model and rolling mechanism. Various prediction models (multiple layer perceptron, chemical model, regression model, and physical model) were used by Jiang et al. (Jiang et al. 2004) to predict the air pollutants. Based on the prediction results, a suitable and effective model among the various models was established. Ni et al. (2017) had used various statistical models for predicting the PM2.5 level on the Beijing PM2.5 dataset. The authors had also claimed that LR models were better in some cases as compared to other models.

Several prediction models had been proposed in the literature. Most of them were dealing with small datasets, and only a few were capable of large datasets. However, there is a need for such prediction models that will efficiently predict the AQI levels and are well suited for the large as well as small datasets, which will also be helpful in developing a new solution for health informatics.

3 Materials and methods

In this section, we are going to present the methods and materials, which have been used for the result findings. This section is subdivided into four subsections, i.e., dataset description, proposed methodology, model comparison, and statistical analysis. In Sect. 1, the comprehensive discussion about the CPCB dataset has been conferred. In Sect. 2, a detailed discussion about the proposed RFERF model has been drawn. In Sect. 3, a concise mathematical ground-based discussion of the seven-prediction model has been impersonated. In Sect. 4, the metrics used in performance evaluation have been addressed.

3.1 Data

For this research work, we had gone through studies of the various polluted cities of India and found Delhi, which is the most polluted city to date in India. This city is heavily polluted because it is India's capital and an industrial hub of the country. Thus, it is a highly dense city in reference to the population. Resultant, the pollution level has been increased due to industrial wastage, bio wastage, and pollution caused by vehicles. The high emanation of various gases like NO, NO_2, NH_3, CO_2, CO, and O_3 is making the AQI and NOx levels high (De Vito et al. 2008, 2012). The toxic particles and elements are soluble in the air, and the presence of these toxic particles makes the air polluted. High pollution levels may have a severe impact on human health. The bad air quality may cause various diseases and increase individuals (Man et al. 2005; Boningari and Smirniotis 2016).

For the simulation, we have extracted the sensor-based data of Delhi from the Central Pollution Control Board (CPCB) of India (CPCB 2020). Various sensor-based devices are placed at 37 different stations to cover the whole Delhi region. These locations are presented in Fig. 1. The green landmark shows the active stations, the yellow landmark

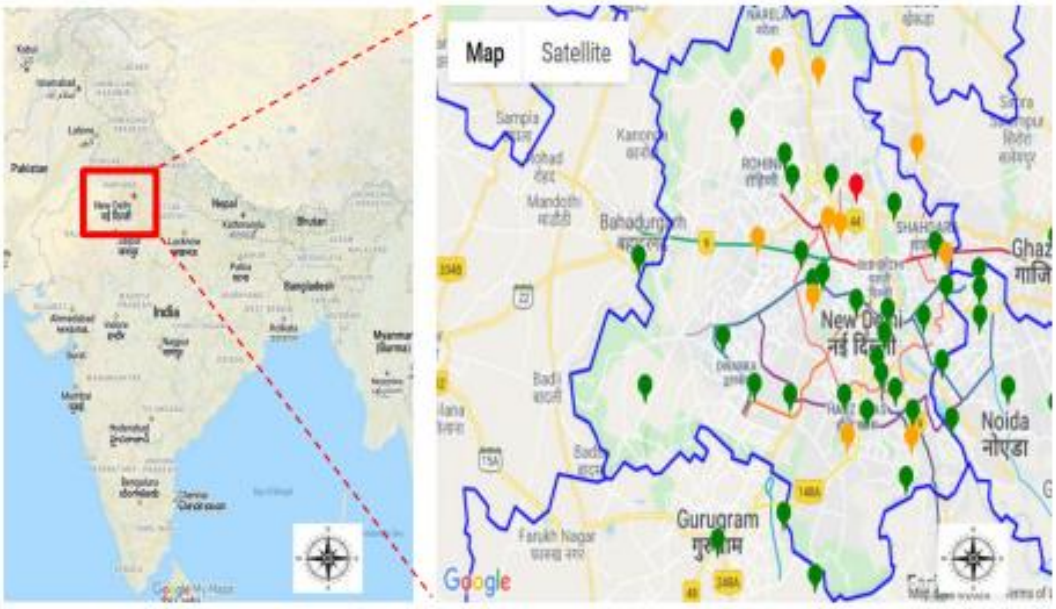


Fig. 1 Base Stations for air quality data collection in the Delhi region: **a** Map of India and **b** Map of Delhi with base stations

denotes the partially operating stations and the red landmark shows the deactivated stations. These stations generate the data with the per hour frequency, which means the data are recorded on an hourly basis. The data from the timespan of January 1, 2019, to January 1, 2021, have been taken for the simulation (Active station only). CPCB dataset contains several liable factors such as (PM10), fine particulate matter (PM2.5), nitrogen monoxide (NO), nitrogen oxide (NOx), nitrogen dioxide (NO_2), ammonia (NH_3), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3). These liable factors can play a vital role in the prediction of AQI and NOx levels. This dataset contains 665,760 rows, and 10 columns or each base station includes 8760 rows and 10 columns (37 base stations have been taken into consideration).

Table 1 describes the various significant features of the dataset, which have been used to predict AQI and NOx levels. The significant features of the dataset have been explored based on multiple parameters such as variable name and its abbreviation, unit of variable, variable type, the period of sample collection, dataset source, and data type.

Table 2 represents the statistical summaries of the dataset based on various significant features. The statistical representation of the various feature contains the variable units, standard prescribed range of variable, the actual range of variable, mean, and standard deviation of each feature. These significant features are playing a key role in the prediction of AQI and NOx.

In Fig. 2, the box plot representation of the various significant features of the dataset has been shown. The features shown in the figure contain various information such as the starting range to the maximum range, the overall distribution of each feature, and the mean of all the features. This information will help us in understanding the data, and based on this; we can perform further investigation.

Table 1 Significant features of the dataset a quick look

Variable	Variable Abbreviation	Unit	Variable type	Collection period	Source	Data type
Particulate Matter10	PM10	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Particulate Matter2.5	PM2.5	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Nitrogen Monoxide	NO	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Nitrogen Oxide	NOx	Ppb	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Nitrogen Dioxide	NO_2	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Ammonia	NH_3	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Sulfur Dioxide	SO_2	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Carbon Monoxide	CO	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Ozone	O_3	$\mu g/m^3$	Pollutant	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous
Air Quality Index	AQI	$\mu g/m^3$	Null	01/01/2019 to 01/01/2021	CPCB (INDIA)	Continuous

Table 2 Variable description

Variable	Unit	Prescribe range		Actual range		Mean	SD
		Min	Max	Min	Max		
PM10	$\mu g/m^3$	0	100	0.14	1000	208.869	154.392
PM2.5	$\mu g/m^3$	0	60	0.7	989.58	106.398	99.803
NO	$\mu g/m^3$	0	200	0.01	499.1	30.339	55.716
NOx	Ppb	0	200	0.01	500	51.994	60.044
NO_2	$\mu g/m^3$	0	200	0.01	485.85	43.873	33.533
NH_3	$\mu g/m^3$	0	200	0.01	494.11	35.515	20.61
SO_2	$\mu g/m^3$	0	80	0.01	194.9	14.821	11.381
CO	$\mu g/m^3$	0	4	0.01	40.25	1.362	1.082
O_3	$\mu g/m^3$	0	180	0.01	997	41.407	59.011
AQI	$\mu g/m^3$	0	100	8.85	1000	217.321	152.63

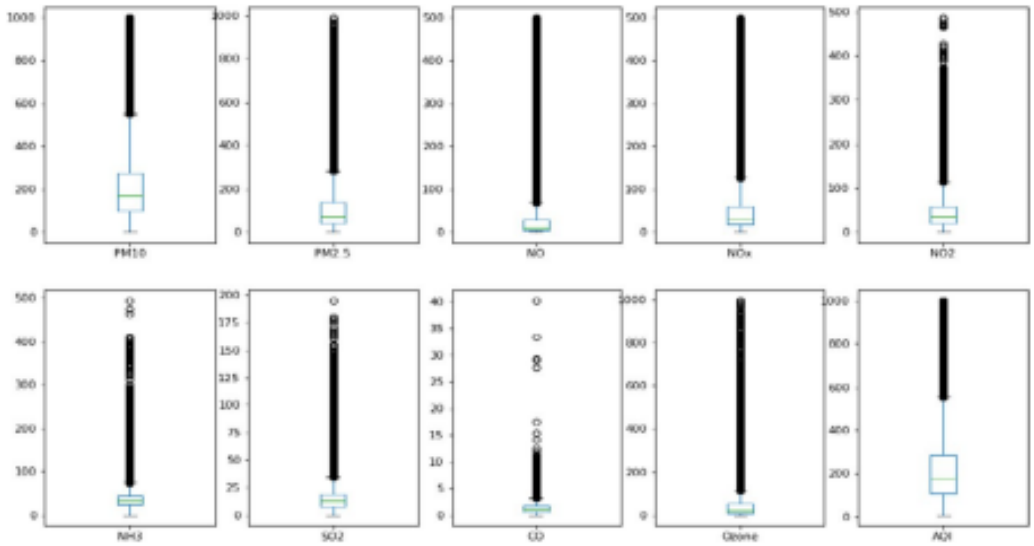


Fig. 2 Box plot of various features in the dataset

3.2 Proposed model

From time to time, various researchers had been successfully applied several machine learning models for long and short-term prediction of air quality. Thus, we started testing various hyper-tuned machine learning algorithms empirically over various AQI datasets and found that the performance of the RFR algorithm is much better as compared to other algorithms. After several trials, we found that our proposed Linear Regression based Recursive Feature Elimination with Random Forest Regression model (RFERF) is better than the traditional forecasting model. The flowchart of the proposed model is shown in Fig. 3. The proposed algorithm contains various steps. In the first step, the CPCB air quality dataset is provided as input. In step two, the data cleaning process on the CPCB dataset is performed. Data cleaning is the process of dropping the missing values and unusual objects (i.e., nothing but some features which are not common in all the stations). In the third step, the pre-processed data are served as an input

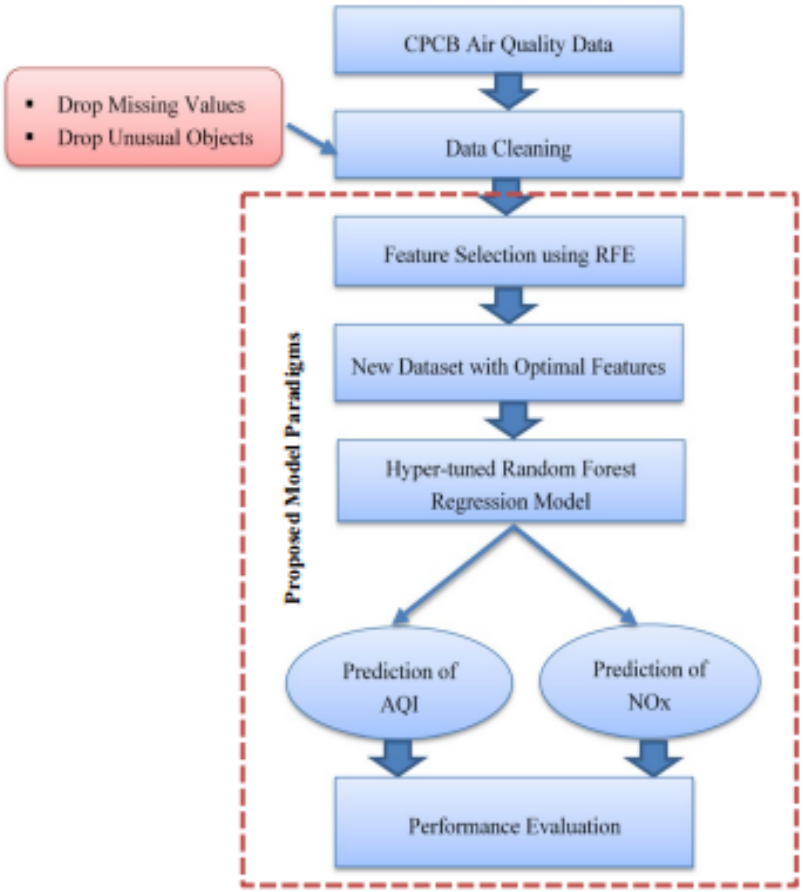


Fig. 3 Flowchart of the proposed model

to the recursive feature elimination (RFE) model for the selection of the best features based on their features ranking. Based on the RFE model evaluation the optimal feature is selected, which forms a new dataset with optimal features. In step four, this new dataset is served as an input to the hyper-tuned RFR model with the tenfold class validation policy for the prediction of AQI and NOx levels. At last, the proposed model's performance is evaluated by using various performance measures [i.e., MAPE, MAE, MSE, RMSE, and Coefficient of Determination ($R2$ score)].

3.2.1 Feature selection using recursive feature elimination model

Recursive feature elimination (RFE) is one of the vital feature selection techniques that has been used to eliminate the weakest feature (or features) from the given dataset, or in other words, we can say that it is used to find the optimal features in the dataset (Granitto et al. 2006; Yan and Zhang 2015). The elimination of the features is continued until the desired number of features will not come. The ranking of the feature is evaluated with the help of feature importance or by finding the correlation coefficient of the data. The primary objective of the REF model is to eliminate the collinearity and dependencies in the model by recursive elimination of the features per loop. The pseudocode of the REF algorithm to find the optimal features in the CPCB dataset has been represented in Algorithm 1.

Algorithm 2. Pseudocode of the Hyper-tuned Random Forest Regression Model

Inputs:
I – *Input dataset (With the Optimal Feature F calculated by RFE Algorithm)*
T – *Testing samples*
$Hyperparameters - (bootstrap, \max\_depth, \max\_feature, \min\_sample\_leaf, min\_sample\_spit, n\_estimators)$

1. **Function** RandomizedSearchCV ($bootstrap, \max\_depth, \max\_feature, \min\_sample\_leaf, min\_sample\_spit, n\_estimators$)
2. Apply randomized search technique on hyperparameter
3. **Return** superlative hyperparameter estimator
4. **Function** RandomForestRegressor($I\_train, T\_train, I\_test, T\_test$)
5. **Use** RandomizedSearchCV for getting the best hyperparameter estimator
6. Fit the model with I_train, T_train
7. Now predict the labels for the I_test
8. **Return** prediction result

Output:
Prediction results with the best hyperparameter estimator.

3.3 Model Comparison

In this section, a concise mathematical grounds discussion of the prediction model utilized in the preliminary evaluation has been impersonated.

3.3.1 Linear Regression (LR)

Linear regression is one of the simple regression models which is most commonly used in predictive analysis. The two variables are being used in the LR model; the first one is an explanatory or descriptive variable. The second one is said to be a dependent or reliant variable. The LR model is used to establish the relationship between the two variables. You can identify the LR equation as the slope formulary (Weisberg 2005).

In Eq. (1) the LR model has been shown, where y is the dependent or reliant variable (which is on the y-axis), x is the explanatory or descriptive variable (which is on the x-axis), a denotes the y-intercept, and b represents the slope in the line.

The mathematical formulation of the LR equation is defined by:

$$y = a + bx \tag{1}$$

The value of a and b are calculated by using the following formulas:

$$b(\text{slope}) = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2} \tag{2}$$

$$a(\text{intercept}) = \frac{n\sum y - b(\sum x)}{n} \tag{3}$$

3.3.2 Gradient Boosting Regression (GBR)

Gradient Boosting Regression (GBR) is one of the ensembled learning-based regression models, which is most commonly used in the predictive analysis (Friedman 2001). The mathematical implementation of the GBR model is delivered in Eq. (4).

$$G_m(x) = \sum_{j=1}^{J} b_{jm} E(x \in R_{jm}) \tag{4}$$

where $E(x \in R_{jm}) = \begin{cases} 1, & if x \in R_{jm} \\ 0, & otherwise \end{cases}$, where b_{jm} denotes the total number of instances resting in region j and R_{jm} indicates the sum of the gradient in region j.

In the next Eq. (5), the regression tree is used for replacing $G_m(x)$ in the GBR method.

$$f_m(x) = f_{m-1}(x) + p_{jm} G_m(x) \tag{5}$$

$$p_m = \text{argmin}_p \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + p_{jm} G_m(x)) \tag{6}$$

Now we get Eq. (8) by using Eq. (7)

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} p_{jm} b_{jm} E(x \in R_{jm}) \tag{7}$$

$$p_m = \text{argmin}_p \sum_{i=1}^{n} L\left(y_i, f_{m-1}(x_i) + \sum_{j=1}^{J} p_m b_{jm} E(x \in R_{jm}) \right) \tag{8}$$

Each region is separated by p_{jm} and R_{jm} so that, b_{jm} can be discarded; thus, the model is updated by using Eqs. (7) and (8).

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} p_{jm} E(x \in R_{jm}) \tag{9}$$

$$p_m = \text{argmin}_p \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \sum_{j=1}^{J} p_{jm} E(x \in R_{jm}) \right) \tag{10}$$

By updating Eq. (7) the overfitting can be prevented

$$f_m(x) = f_{m-1}(x) + J \cdot \sum_{j=1}^{J} p_{jm} E(x \in R_{jm}) \tag{11}$$

3.3.3 Decision Tree Regression (DTR)

Decision Tree Regression (DTR) is one of the tree-based regression models, most commonly used in predictive analysis (Tso and Yau 2007). The prediction process is achieved by calculating the serval errors with the help of the below-discussed equations. In Eq. (12), the mathematical formulation of mean square error is given.

$$E(p) = \frac{1}{N}\sum_{i=1}^{N}(u_i - p(v_i))^2 \tag{12}$$

Where N is the sample size, (x_i, v_i) is the learning sample and value of $i = 1, 2, \ldots N$.

A validation error is calculated using Eq. (12)

$$E^{cv}(p) = \frac{1}{N_k}\sum_{k}\sum_{(x_{i,}v_i)\in x_k}(v_i - p^{(k)}(x_i))^2 \tag{13}$$

where the value of $p^{(k)}$ is calculated by subtracting the x to x_k, i.e., $x - x_k$ (where x is a sample).

From Eq. (14) the test sample error is calculated.

$$E^{ts}(p) = \frac{1}{N_2}\sum_{(x_{i,}v_i)\in x_2}^{N}(v_i - p(u_i))^2 \tag{14}$$

$$R(t) = \frac{1}{N_w(t)}\sum_{i\in t} w_i f_i (x_i - \overline{v}(t))^2 \tag{15}$$

where $N_w(t)$ is the Weighted number of cases in the node (t), w_i is the value of the weighted variable, v_i is the response variable and $\overline{v}(t)$ is the weighted mean for the node (t).

3.3.4 Multi-Layer Perceptron Regression (MLP)

Multi-Layer Perceptron Regression (MLP) is one of the artificial neural network (ANN) based feed-forward regression models, which is most commonly used in the predictive analysis (Agirre-Basurko et al. 2006). The MLP model contains three layers, such as the input layer, hidden layer, and output layer. In the MLP model, there may be many hidden layers, and each hidden layer is calculated with the activation function shown in Eq. (16).

$$A_k = \varnothing_0\left(\alpha_k + \sum_{j\to k}\omega_{jk}\varnothing_h\left(\alpha_j + \sum_{i\to j}\omega_{ij}x_i\right)\right), \tag{16}$$

where $(\varnothing_h)$ is the hidden layer tangent hyperbolic function, $(\varnothing_0)$ is the output layer exponential function, x_i denotes the input and A_k denotes the output.

Penalized regularization technique, which is an optimization criterion, is used to minimize and optimize the penalized log in the model. Equation 17 is used to minimize the penalize log.

$$L = -\text{loglikelihood} + \lambda\sum_{\text{weights}}\omega_{ij}^2 \tag{17}$$

where λ is the cross-validation parameter.

3.3.5 K-Nearest Neighbors Regression (KNNR)

The KNNR is one of the nonparametric models, which is most commonly used in the predictive analysis (Devroye et al. 1994). The KNNR Model consists of the state vector and distance vector. The value of k denotes the number of nearest neighbors. The coefficient of weight is calculated using Eq. (18). The weight should be correlated with the distance between the time gap and the forecasting scenario.

$$D_i = \sqrt{\sum_j w_j (V_j - v_{ji})^2} \tag{18}$$

where D_i is the distance group based on current and historical data, w_j is the weight state vector, V_j is the value of state vector and v_{ji} is the value of state vector j and i is the historical database.

In Eq. (19), the next period for predicting values that are based on the feature is described.

$$S_m(t+1) = \sum_{g=1}^{k} \frac{d_i^{-1}}{\sum_{g=1}^{k} d_i^{-1}} S_{g,h}(t+1), \tag{19}$$

where $S_m(t+1)$ is average feature selection time, $(t+1)$ is the time involved in searching nearest neighbor, $S_{gh}(t+1)$ is stands for the average speed of the nearest neighbor, and k denotes the number of the nearest neighbor.

3.3.6 Support vector regression (SVR)

The SVR is a simple regression model most commonly used in predictive analysis. SVR model is used to predict the data based on the blow described Eq. (20) (Drucker et al. 1997).

$$f(x) = \omega \varnothing(x) + b \tag{20}$$

$$\varnothing : R^n \to F, \omega \epsilon F$$

where ω and b are the coefficient of $\varnothing(x)$ and $\varnothing$ denotes the high dimensional feature space.

$$R_{\mathrm{SVR}}(c) = R_{\mathrm{emp}} + \frac{1}{2}\|\omega\|^2 = c \times \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon \left(d_i, y_i\right) + \frac{1}{2}\|\omega\|^2 \tag{21}$$

where R_{emp} denotes the empirical risk, $\frac{1}{2}\|\omega\|^2$ represent the Euclidian norms, $c \times \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon \left(d_i, y_i\right)$ denotes empirical error and c denotes the cost associated with empirical risk

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon & |d - y| \geq \varepsilon \\ 0 & otherwise \end{cases} \tag{22}$$

The loss function is represented in Eq. (22) and can be obtained by using Eq. (21). Minimize,

$$R_{SVR}(\omega, \xi^*) = \frac{1}{2}\|\omega\|^2 + c\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{23}$$

$$d_i - \omega\varnothing(x_i) - b_i \leq \varepsilon + \xi_i$$

Subjected to,

$$\omega\varnothing(x_i) + b_i - d_i \leq \varepsilon + \xi^*\xi^* \geq 0$$

Equation (23) describes the minimization of a testing error where ξ^* and ξ_i^* denotes the slack variable for the measurement of variable ups and downsides, respectively.

3.3.7 Random Forest Regression (RF)

Random Forest Regression is one of the ensemble learning-based regression models, which is most commonly used in predictive analysis. This regression was developed by TinKam Ho in 1995 (Liaw and Wiener 2002). It performed the task with the help of a decision tree where the tree was constructed by using algorithm $h(x;\theta_k), k = 1, \ldots k$. The x and k represent the input vectors, p represents the length of random vector x, and θ_k are an independent variable that distributes the dataset into the training set T and a random feature variable of θ (where θ is a sample of some distribution).

The mathematical implementation of the RF model is delivered in Eq. (24).

$$\overline{h}(X) = (1/x)\sum_{k=1}^{K} h(x;\theta_k) \tag{24}$$

where $k \to \infty$ ensures a large number of predictions.

$$E_{x,y}(Y - \overline{h}(x))^2 \to E_{x,y}(Y - E_\theta h(X;\theta))^2 \tag{25}$$

Equation 26 describe the prediction error of random forest and the designated pE_t^* is implemented in Eq. 25 to protect the overfit of the random forest model. Now we predict the average error of an individual tree $h(X;\theta)$, which is described in Eq. (26)

$$pE_t^* = E_\theta E_{x,y}(Y - h(x;\theta))^2 \tag{26}$$

Here we assume that all θ are unbiased for every tree, i.e.,

$$EY = E_x h(x;\theta)$$

Then,

$$PE_f^* \leq \overline{p}PE_t^* \tag{27}$$

where $\overline{p}$ is the weighted correlation between residuals $y - h(x;\theta)$and$y - h(x;\theta')$ for independent of θ and θ'.

3.4 Performance evaluation measures

Various performance evaluation metrics, i.e., MAPE, MAE, MSE, RMSE, and Coefficient of Determination (*R*2 score) have been used to measure the performance, accuracy, and suitability of prediction models. In this section, the evaluation metrics are described in detail with their mathematical foundation (Nagelkerke 1991).

3.4.1 Mean square error (MSE)

The mean square error (MSE) is one of the vital statistical measures which is used to evaluate the average of squares errors. Where $(\hat{x}_i - x_i)^2$ represents the squares of errors, N denotes the number of errors, $\hat{x}_i$ indicates the observed values and x_i indicates predicted values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2 \tag{28}$$

3.4.2 Mean absolute error (MAE)

The MAE is a critical statistical measure used to evaluate the average of absolute errors. Where $|Y_i - X_i|$ represents the absolute errors, N denotes the number of errors, Y_i indicates actual values, and X_i indicates the predicted values.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|Y_i - X_i| \tag{29}$$

3.4.3 Root mean square error (RMSE)

The root mean absolute error (RMSE) is a standard derivation for residuals. It is one of the vital statistical measures for the validation of prediction results. Residuals mean the error prediction, which measures the distance among the data points and regression line. Where $(\hat{x}_i - x_i)^2$ represents the squares of errors, N denotes the number of errors, $\hat{x}_i$ indicates the observed values and x_i indicates the forecasted values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2} \tag{30}$$

3.4.4 Coefficient of determination (*R*2 score)

*R*2 score stands for Coefficient of Determination, which is one of the crucial statistical measures for the validation of prediction results. It is a division ratio of the first sum of squares of errors (explained variation) by the second sum of squares of errors (unexplained variation), and the value of the *R*2 score is calculated by subtracting the division ratio by one. where $(\hat{x}_i - x_i)^2$ represents the squares of residuals, $(\hat{x}_i - y_i)^2$ represents the squares of the total, N denotes the number of errors, $\hat{x}_i$ indicates the observed values, x_i and y_i indicates the forecasted values.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{x}_i - x_i)^2}{\sum_{i=1}^{N}(\hat{x}_i - y_i)^2} \tag{31}$$

3.4.5 Mean Absolute Percentage Error (MAPE)

The MAPE is an essential statistical measure used to evaluate the accuracy of the forecast model. Where N denotes the number of predicted samples, Y_i indicates actual values, and X_i indicates the predicted values.

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{Y_i - X_i}{Y_i}\right| \times 100\% \tag{32}$$

4 Results

In this section, we are going to address the prediction result of the AQI and NOx based on the eight prediction models, i.e., DTR, GBR, KNNR, LR, Multi-layer Perceptron Regression (MLPR), RFR, SVR, and our proposed Linear Regression based RFERF.

4.1 Best model identification

The identification of the correct prediction model, which will be best suited for the prediction paradigms, is one of the challenging tasks. The primary aim of this research work is to emphasize the prediction of the AQI and Air Pollutant Concentration (NOx) as well as find out the impact of various pollutants on individuals' health too. Forecasting AQI and NOx will also be helpful in giving a roadmap toward the possible solution for minimizing the impact of bad air quality on individuals' health.

For best model identification seven well-established regression models and one proposed hybrid model, i.e., DTR, GBR, KNNR, Linear Regression (LR), MLP Regression (MLPR), RFR, SVR, and Linear Regression based RFERF has been shown. These regression models have been used for the prediction of AQI and NOx levels. All the experimental analysis has been performed using a ten cross-validation policy. The primary aim of this study is to find out the best prediction model among eight regression models that will be well suited and highly accurate.

The best hyper-parameter setting, which is used in the preliminary evaluation, is shown in Table 3. This table consists of information such as the name of the prediction model, their hyperparameters, their parameters selection, and the best hyperparameter taken in the preliminary evaluation. Prior to conducting the preliminary evaluation, each model is hyper-tuned under several selection criteria to obtain the fittest hyperparameter for the prediction model.

4.2 Performance evaluation of AQI prediction model

In Table 4, the result of eight regression models based on the five performance evaluation measures for the prediction of AQI on the CPCB dataset of the Delhi region has been

Table 3 Hyperparameter selection

Prediction model	Hyperparameter	Parameter selection	Best hyper-parameter used
DTR	max_depth	[3, None]	3
	Criterion	[gini, entropy]	gini
	min_samples_leaf	[1,3,5,7, 9]	5
	max_features	[1,3,5,7, 9]	5
GBR	learning_rate	[0.001,0.002,0.005]	0.005
	n_estimators	[20, 40, 60, 80, 100, 120, 140, 160, 180, 200]	160
	max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]	110
	min_samples_split	[2, 5, 10]	5
	min_samples_leaf	[1, 2, 3, 4]	3
	max_features	[auto, sqrt]	auto
KNNR	n_neighbors	[1, 2, 5, 7, 10, 15, 18, 21]	1
	metric	[Euclidean, Manhattan, Minkowski]	Minkowski
	Weight	[Uniform, Distance]	Uniform
LR	Dual	[True, False]	False
	max_iter	[100,110,120,130,140]	100
	C	[1,1.5,2,2.5]	2
MLPR	hidden_layer_sizes	[(10,), (20,)]	(20,)
	Activation	[tanh, relu]	relu
	Solver	[sgd, adam]	adam
	Alpha	[0.0001, 0.05]	0.0001
	learning_rate	[constant, adaptive]	constant
RFR	Bootstrap	[True, False]	True
	max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]	70
	max_features	[auto, sqrt]	auto
	min_samples_leaf	[1, 2, 3, 4]	4
	min_samples_split	[2, 5, 10]	10
	n_estimators	[20, 40, 60, 80, 100, 120, 140, 160, 180, 200]	80
SVR	Kernel	rbf	rbf
	C	[0.1, 1, 100, 1000]	100
	Gamma	[0.0001, 0.001,0.01 0.005, 0.1, 1, 3, 5]	0.1
	Epsilon	[0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]	0.0001
RFERF	Bootstrap	[True, False]	True
	max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]	100
	max_features	[auto, sqrt]	auto
	min_samples_leaf	[1, 2, 3, 4]	2
	min_samples_split	[2, 5, 10]	2
	n_estimators	[20, 40, 60, 80, 100, 120, 140, 160, 180, 200]	20

Table 4 Performance evaluation of AQI prediction model

Algorithms	Accuracy	MAPE	MAE	MSE	RMSE	*R*2 score
DTR	94.27	5.73	7.34	903.16	30.05	0.96
GBR	92.74	7.26	8.99	532.12	23.07	0.98
KNNR	94.44	5.56	7.93	517.75	22.75	0.98
LR	90.37	9.63	14.93	885.56	29.76	0.96
MLPR	94.07	5.93	7.50	448.78	21.18	0.98
RFR	94.98	5.02	6.53	461.54	21.48	0.98
SVR	95.26	4.74	7.43	708.78	26.62	0.97
RFERF	99.82	0.18	0.37	22.59	4.75	1.00

Table 5 Performance evaluation of NOx prediction model

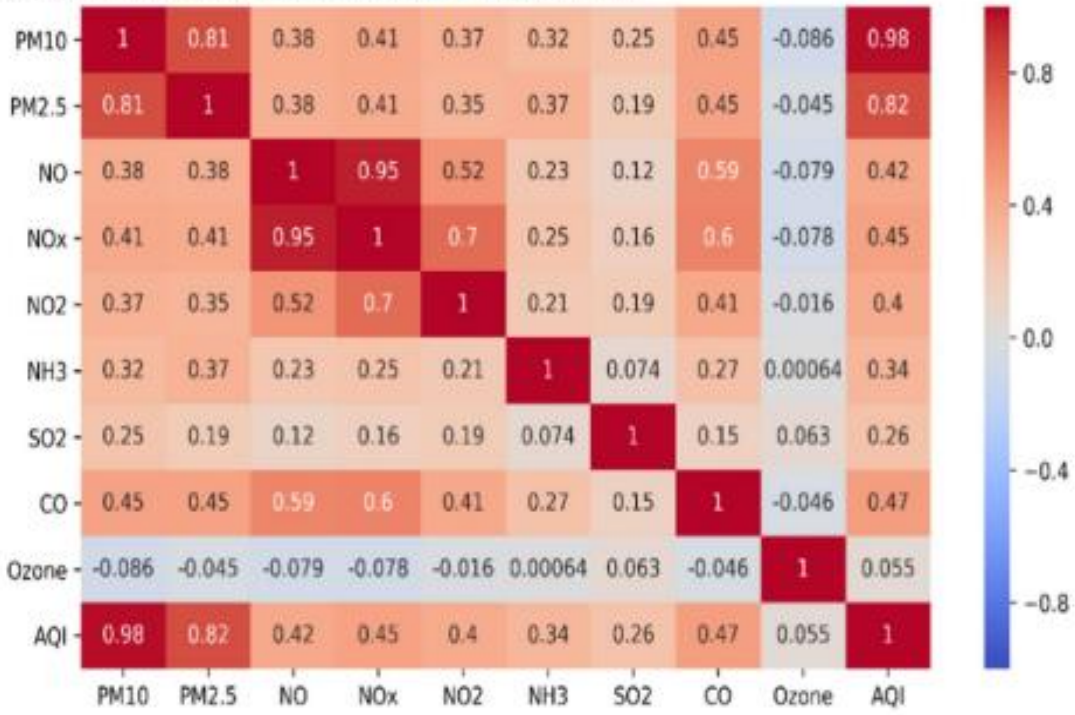
Algorithms	Accuracy	MAPE	MAE	MSE	RMSE	*R*2 score
DTR	88.15	11.85	5.09	180.11	13.42	0.95
GBR	85.53	14.47	5.78	121.29	11.01	0.97
KNNR	82.72	17.28	6.08	127.67	11.30	0.96
LR	82.22	17.78	6.49	139.47	11.81	0.96
MLPR	87.05	12.95	5.11	99.67	9.98	0.97
RFR	89.45	10.55	4.45	94.23	9.70	0.97
SVR	88.73	11.27	5.05	180.70	13.44	0.95
RFERF	91.23	8.77	3.92	81.46	9.03	0.98

shown. For the validation of the predicted result accuracy should be high, the MAPE ought to be low, MAE ought to be low, MSE ought to be low, RMSE ought to be low and *R*2 score ought to be high but less than 1. Based on these validation measures, we have evaluated the performance of each model to find out the best prediction model among all the regression models. Among all the eight prediction models, the proposed RFERF model wins the battle with the highest prediction accuracy of 99.82, lowest MAPE at 0.18, lowest MAE at 0.37, lowest MSE at 22.59, lowest RMSE at 4.75, and highest *R*2 score 1.00.

4.3 Performance evaluation of NOx prediction model

Table 5 shows the result of eight regression models based on the five performance evaluation measures for the prediction of NOx on the CPCB dataset of the Delhi region. For the validation of the predicted result accuracy should be high, the MAPE ought to be low, MAE ought to be low, MSE ought to be low, RMSE ought to be low and *R*2 score ought to be high but less than 1. Based on these validation measures, we evaluated each model's performance to find out the best prediction model among all the regression models. Among all the eight prediction models, the proposed RFERF model wins the battle with the highest prediction accuracy of 91.23, lowest MAPE at 8.77, lowest MAE at 3.92, lowest MSE at 81.46, lowest RMSE at 9.03, and highest *R*2 score 0.98.

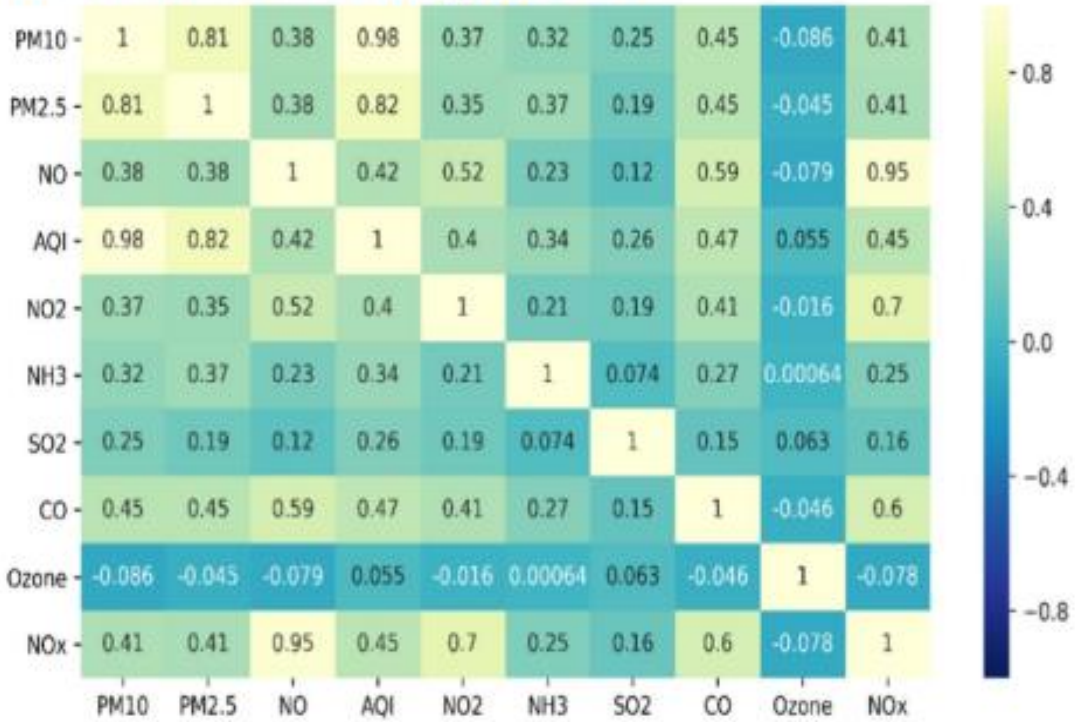
Table 6 Correlation coefficient matrix for air pollutants of AQI



5 Discussion

There are many associated factors that are playing a vital role in affecting air quality. Directly or indirectly, the participation of these factors is responsible for the emission of various pollutants. These pollutants are soluble in the air, and they are very harmful to human health too. The poor diffusion condition may lead to a drastic increment in the

Table 7 Correlation coefficient matrix for air pollutants of NOx



concentration level of various pollutants. Due to this, massive changes in the level of the AQI and Air Pollutant Concentration (NOx) have been seen. The diffusion is nothing but the movement of the air particle from the high concentration area to the low concentration area. The initial pre-processing of data is done by dropping the unusual object and missing values. The dataset is consist of various liable factors such as (PM10), fine particulate matter (PM2.5), nitrogen monoxide (NO), nitrogen oxide (NOx), nitrogen dioxide (NO_2), ammonia (NH_3), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3). Before sending the data to the regression model, we have calculated the correlation between the various liable factors or pollutants.

In Tables 6 and 7, the correlation among the ten air pollutants has been shown for AQI and NOx, respectively. From the correlation, we found that most of the indicators are highly correlated. Hence, in the absence of one or more air pollutant indicators, the remaining indicators can be used to predict the missing indicator. The prediction of the AQI and Air Pollutant Concentration (NOx) has been taken as examples for erecting the regression model and to verify the viability of the proposed regression model.

5.1 Air Quality Index (AQI) prediction of Delhi

For the experimental analysis, we have taken the AQI of Delhi as the regression objective. The data from 01/01/2019 to 01/01/2021, of the CPCB dataset, have been used for training and testing purposes with a ratio of seventy and thirty, respectively. Seventy percent of the dataset has been used for training the model, and the rest thirty percent has been used for testing purposes.

The regression hyperparameter tuning has been performed for all the eight prediction models (i.e., DTR, GBR, KNNR, Linear Regression, MLP Regression, RFR, SVR, and proposed hybrid model), and based on that; the best regression parameter has been used for training the model.

As presented in Fig. 4, the x-axis denotes the order of testing samples, and the y-axis indicates the target values (AQI). The estimation of the testing samples has been evaluated by using DTR, GBR, KNNR, LR, MLPR, RFR, SVR, and RFERF regression models. All the prediction results and actual results are represented in Fig. 4a to h with subplots. In the subplot, the line chart has been used to show a comparison between the actual values (real values) and the predicted values (observed values). All the models emulate the prediction of the AQI relatively well. In the context of performance, the RFERF model performs better among all the regression models.

5.2 Air Pollutant Concentration (NOx) prediction of Delhi

For the experimental analysis, we have taken the Air Pollutant Concentration (NOx) of Delhi as the regression objective. The data from 01/01/2019 to 01/01/2021, of the CPCB dataset, have been used for training and testing purposes with a ratio of seventy and thirty, respectively. Seventy percent of the dataset has been used for training the model, and the rest thirty percent has been used for testing purposes.

The regression hyperparameter tuning has been performed for all the eight prediction models (i.e., DTR, GBR, KNNR, LR, MLPR, RFR, SVR, and the proposed RFERF model), and based on that; the best regression parameter has been used for training the model.

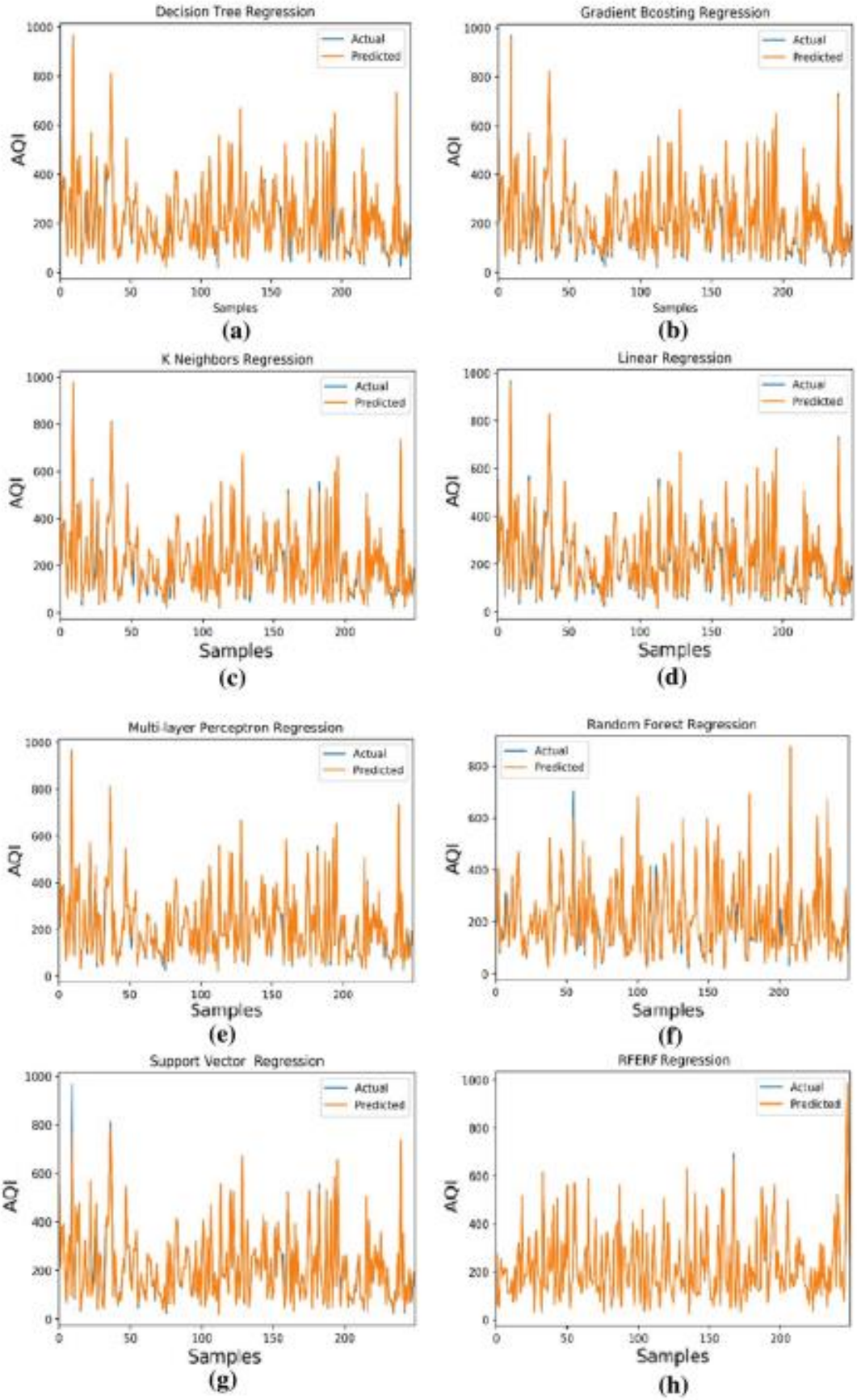


Fig. 4 Scatterplots of the AQI predictions (predicted values over actual values): **a** DTR model, **b** GBR model, **c** k-NR model, **d** LR model, **e** MLPR model, **f** RFR model, **g** SVR model, **h** RFERF model

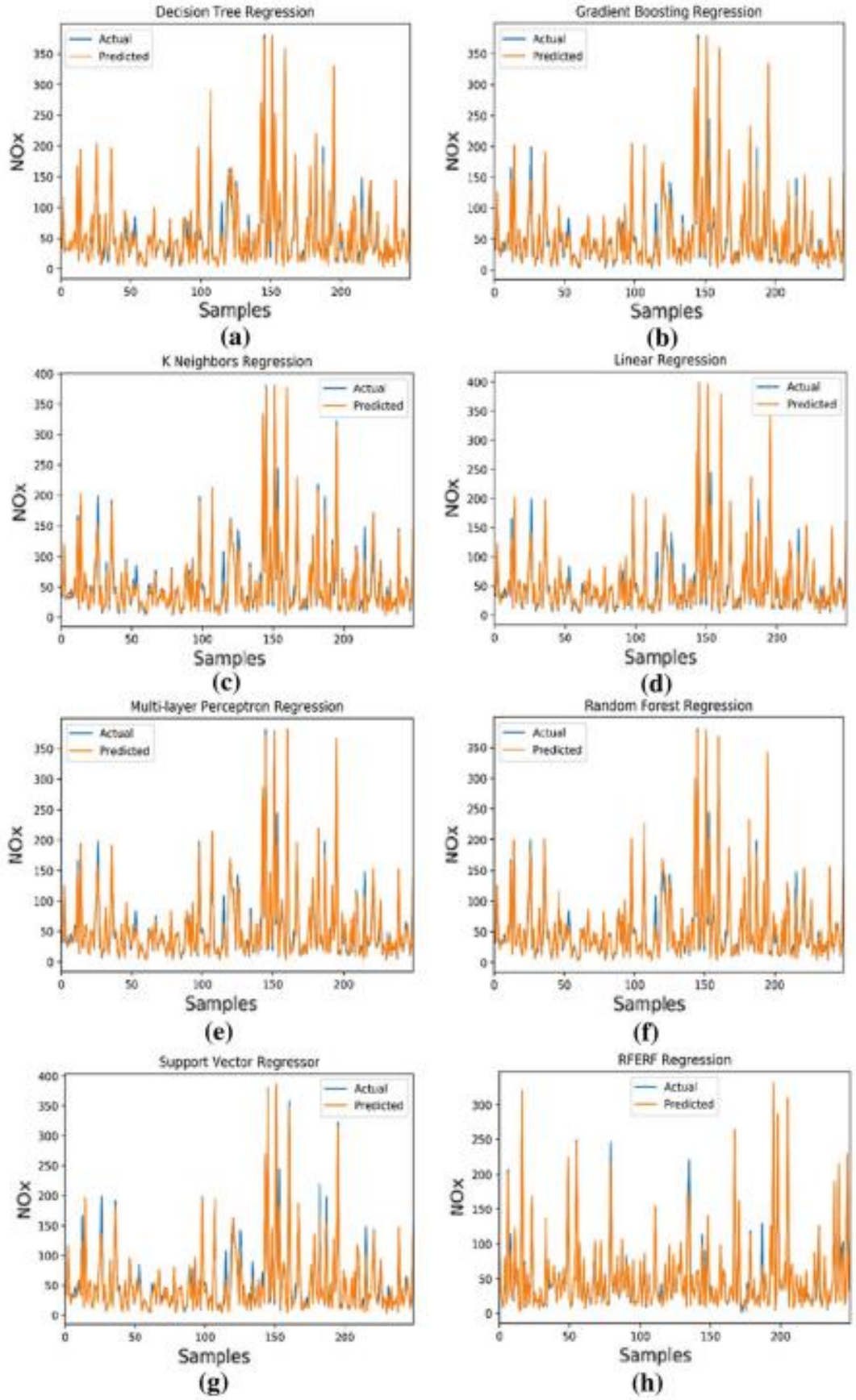


Fig. 5 Scatterplots of the NOx predictions (predicted values over actual values): **a** DTR model, **b** GBR model, **c** k-NR model, **d** LR model, **e** MLPR model, **f** RFR model, **g** SVR model, **h** RFERF model

As presented in Fig. 5, the *x*-axis denotes the order of testing samples, and the y-axis denotes the target values (NOx). The testing samples' estimation has been evaluated using DTR, GBR, KNNR, LR, MLPR, RFR, SVR, and RFERF regression models. All the prediction results and actual results are represented in Fig. 5a to h with subplots. In the subplot, the line chart has been used to show a comparison between the actual values (real values) and the predicted values (observed values). All the models emulate the prediction of the Air Pollutant Concentration (NOx) relatively well. In the context of performance, the RFERF model performs better among all the regression models.

5.3 Effect on health

The effect of a high AQI level on a person's health is presented in Table 8 (Ketu and Mishra 2021d, e; Ruggieri and Plaia 2012; Friedman 1996; Cleland and Ginneken 1988; Anderson et al. 2012). We have divided the various effect into three subsections. The first section shows the short-term impact of the high level of AQI on individuals' health and also defines the various diseases which are caused by it. In the second section, the long-term effect of the level of AQI on human health with associated diseases has been shown. In the last section, the persons who are already dealing with some curial diseases, and need extensive care have been discussed.

The effect of various pollutants on individuals' health is presented in Table 9 (Ruggieri and Plaia 2012; Friedman 1996; Cleland and Ginneken 1988; Anderson et al. 2012; Ketu and Mishra 2021d, 2022b). We divided the table into two-part; in the left-most column, various pollutants were taken, and in the right column, their effect on health was described. Most of the pollutants affect the person's respiratory system, cardiovascular or circulatory system, excretory system (kidney or urinary), endocrine system, nervous system, circulatory system, lymphatic system, digestive system, ophthalmic system, and integumentary system (skin).

5.4 Possible solutions

In this section, a brief discussion about the possible solutions by linking machine learning with sensor-generated AQI data for air quality prediction to reduce air pollution levels which will be an adequate and appropriate way to solve some related environmental glitches. This section also talks about how we can minimize the impact of pollution levels on individuals' health (Ketu and Mishra 2021b, d, 2022d).

5.4.1 Embed real-time AQI level with the Google Map

We live in the technological era and are familiar with the latest technologies. The new generation of smart gadgets has become an essential part of our daily routine. So, the best way to convey the information about the current, as well as upcoming (i.e., by linking machine learning with sensor-generated AQI data for air quality prediction) air quality values will be via Google Maps. It will not only give real and future time information about AQI levels but also give the present and future traffic conditions. Based on the AQI level, it will also

Table 8 The effect of high AQI level on a person's health

Effect on health		
Short term	Long term	Severe health problems for
(a) Serious cardiovascular illness	(a) Faster aging of the lungs	(a) The person suffering from heart disease
(b) Serious respiratory illness	(b) Reduction in lung capacity	(b) The person suffering from congestive heart failure
(c) Cause more strain on the lungs and heart	(c) Reduction in lungs functionality	(b) The person suffering from coronary artery syndrome
(d) Damaged respiratory system cells	(d) Bronchitis	(d) The person suffering from asthma,
	(e) Asthma	(e) The person suffering from Emphysema
	(f) Possibly cancer	(f) The person suffering from COPD (Chronic Obstructive Pulmonary Disease)
	(g) Emphysema	(g) Women with pregnancy
	(h) Shorter life span	(h) Outdoor labors
		(i) Old age people and children below 14 years of age
		(j) Sportspersons who exercise strongly in outdoors

Table 9 Effect various pollutants on a person's health

Pollutants	Effect on health
PM10	Precocious death due to lung or heart disease Non-life-threatening heart attacks Irregular heartbeat Serious asthma Reduction in lung function Impact on respiratory system-related diseases like coughing, irritation, blockage in the airways, or trouble breathing
PM2.5	Precocious death Cardiac arrhythmias Heart attacks Asthma attacks Bronchitis
NO	Irritation in the eyes, respiratory system, and skin Serious asthma Choking and coughing Headache Nausea Trouble in breathing Abdominal pain
NOx	Irritation in the eyes, respiratory system, and skin Serious asthma Choking and coughing Headache Nausea Trouble in breathing Abdominal pain
NO_2	Irritation in the eyes, respiratory system, and skin Serious asthma Choking and coughing Headache Nausea Trouble in breathing Abdominal pain
NH_3	Instant burning in the throat, nose, and breathing tract Bronchiolar and alveolar edema Destruction in airway Respirational distress failure Irritation in the throat and nose Coughing Difficulty in the olfactory nerve
SO_2	Wheezing Shortness in breathing Tightness in chest A severe problem in the respirational system Reduction in lungs functionality Severe breathing difficulty in asthma patient Discomfort in the breathing tract
CO	Severe headache Faintness, nausea, and vomiting Unconsciousness and precocious death Severe heart disease

Table 9 (continued)

Pollutants	Effect on health
O_3	Chest pain
	Coughing, sore throat, and dry throat
	Inflammation in airway
	Irritation in throat
	Reduction in lung functionality
	Cause damage to lung tissue
	Bronchitis
	Asthma
	Emphysema
	Emphysema
	Wheezing
	Headache and vomiting

suggest the appropriate route in which the pollution level will be minimum. Thus, we can take preventive action to reduce the impacts.

5.4.2 Diversion of heavy vehicle

Linking machine learning with sensor-generated AQI data for air quality prediction will help to improve the current and future traffic conditions. In case of bad air quality prediction, the heavy service vehicles should be restricted to enter the city and in such areas, the delivery of the product must be done with low pollutant vehicles for picking up and dropping the goods. To reduce the pollution and traffic load in high congested areas (i.e., severely polluted areas) flyovers, ring roads and overpasses should be constructed which will not only reduce the pollution level but will also help us in reducing the overhead traffic conditions.

5.4.3 Go green

The greenery and air both are dependent on each other. If we want to maintain good air quality, then we have to maintain the greenery, which means the trees are one of the important resources for maintaining fresh air quality. Trees take carbon dioxide and give us oxygen, which is the basic need of all living creatures. Thus, combining machine learning with sensor-generated AQI data for air quality forecasting will help identify areas that have become highly polluted. The use of cycles or fuel-free vehicles in such contaminated areas will not only maintain the overall quality of the air but also play an essential role in building a healthier environment.

5.4.4 Smart traffic management

For building a healthy environment, traffic plays an important role. So, there is a need for efficient traffic management strategies which can improve the situation of traffic jams (specifically in those bad air quality areas). Due to population growth, the traffic overhead condition is taking place. Resultantly, a massive number of on-road vehicles have been seen. The huge number of on-road vehicles leads to traffic jams. The slow traffic

movement leads to more burning of fuel and which is one of the essential reasons for increasing the pollution level.

6 Conclusion

The accurate forecasting of air quality is one of the essential practical and theoretical significance for the individuals; without it, neither the public nor the government can efficiently evade the damage caused by air pollution to human health. With the help of accurate air quality forecasting, we can improve the capability of the emergency response in the worst pollution scenarios. In this paper, we are focused on developing a mechanistic and quantitative prediction model for the prediction of the AQI and Air Pollutant Concentration (NOx) levels with a clear environmental interpretation. The proposed model is based on the Linear Regression-based RFERF. For the experimental analysis, the seven well-established machine learning models have been taken, and these models are compared with our proposed model to find out their suitability and correctness. For the prediction of AQI and NOx, the data of the CPCB of India has been taken. The proposed model performs superior compared to other prediction models with better accuracy, lower time complexity, higher prediction rate, and suitability for the large dataset. The proposed hybrid RFERF model enriched the prediction accurateness of air indicators and offered supervision for analyzing and modeling the air quality. This work also explains that linking machine learning with sensor-generated AQI data for air quality prediction is an effective and appropriate way to solve some related environmental glitches and help establish the impact of various pollutants on human health with possible solutions.

Funding No funding is received.

Data availability The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Written informed consent for publication was obtained from all participants.

References

Agirre-Basurko E, Ibarra-Berastegi G, Madariaga I (2006) Regression and multilayer perceptron-based models to forecast hourly O_3 and NO_2 levels in the Bilbao area. Environ Model Softw 21(4):430–446

Anderson JO, Thundiyil JG, Stolbach A (2012) Clearing the air: a review of the effects of particulate matter air pollution on human health. J Med Toxicol 8(2):166–175

Athanasiadis IN, Kaburlasos VG, Mitkas PA, Petridis V (2003) Applying machine learning techniques on air quality data for real-time decision support. In: First international NAISO symposium on information technologies in environmental engineering (ITEE'2003), June, Gdansk, Poland

Biancofiore F, Busilacchio M, Verdecchia M, Tomassetti B, Aruffo E, Bianco S, Carlo Colangeli S, Rosatell G, Di Carlo P (2017) Recursive neural network model for analysis and forecast of PM10 and PM2.5. Atmosph Pollut Res 8(4):652–659

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Boningari T, Smirniotis PG (2016) Impact of nitrogen oxides on the environment and human health: Mn-based materials for the NOx abatement. Curr Opin Chem Eng 13:133–141

Cabaneros SMS, Calautit JKS, Hughes BR (2017) Hybrid artificial neural network models for effective prediction and mitigation of urban roadside NO_2 pollution. Energy Procedia 142:3524–3530

Cabaneros SMS, Calautit JK, Hughes BR (2019) A review of artificial neural network models for ambient air pollution prediction. Environ Model Softw 119:285–304

Chelani AB, Rao CC, Phadke KM, Hasan MZ (2002) Formation of an air quality index in India. Int J Environ Stud 59(3):331–342

Chen M, Mao S, Liu Y (2014) Big data: a survey. Mobile Netw Appl 19(2):171–209

Cleland JG, Van Ginneken JK (1988) Maternal education and child survival in developing countries: the search for pathways of influence. Soc Sci Med 27(12):1357–1368

Corani G (2005) Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. Ecol Model 185(2–4):513–529

CPCB (2020) Dataset. https://app.cpcbccr.com/ccr/#/caaqm-dashboard-all/caaqm-landing/data

De Vito S, Massera E, Piga M, Martinotto L, Di Francia G (2008) On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. Sensors Actuators B Chem 129(2):750–757

De Vito S, Piga M, Martinotto L, Di Francia G (2009) CO, NO_2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic Bayesian regularization. Sensors Actuators B Chem 143(1):182–191

De Vito S, Fattoruso G, Pardo M, Tortorella F, Di Francia G (2012) Semi-supervised learning techniques in artificial olfaction: a novel approach to classification problems and drift counteraction. IEEE Sensors J 12(11):3215–3224

Deswal S, Verma V (2016) Annual and seasonal variations in air quality index of the national capital region, India. Int J Environ Ecol Eng 10(10):1000–1005

Devroye L, Gyorfi L, Krzyzak A, Lugosi G (1994) On the strong universal consistency of nearest neighbor regression function estimates. Ann Stat 22(3):1371–1385

Drucker H, Burges CJ, Kaufman L, Smola AJ, Vapnik V (1997) Support vector regression machines. In: NIPS-3: proceedings of the 1990 conference on Advances in neural information processing systems, pp 155–161

Du X, Kong Q, Ge W, Zhang S, Fu L (2010) Characterization of personal exposure concentration of fine particles for adults and children exposed to high ambient concentrations in Beijing, China. J Environ Sci 22(11):1757–1764

Fan S, Hazell PB, Thorat S (1999) Linkages between government spending, growth, and poverty in rural India, vol 110. International Food Policy Research Institute, Washington, DC

Friedman JM (1996) The effects of drugs on the fetus and nursing infant: a handbook for health care professionals. Johns Hopkins University Press, Baltimore

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232

Fu M, Wang W, Le Z, Khorram MS (2015) Prediction of particular matter concentrations by developed feed-forward neural network with rolling mechanism and gray model. Neural Comput Appl 26(8):1789–1797

Fuller GW, Carslaw DC, Lodge HW (2002) An empirical approach for the prediction of daily mean PM10 concentrations. Atmos Environ 36(9):1431–1441

Ganguly ND, Tzanis CG, Philippopoulos K, Deligiorgi D (2019) Analysis of a severe air pollution episode in India during Diwali festival—a nationwide approach. Atmósfera 32(3):225–236

Granitto PM, Furlanello C, Biasioli F, Gasperi F (2006) Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. Chemom Intell Lab Syst 83(2):83–90

Ibarra-Berastegi G, Elias A, Barona A, Saenz J, Ezcurra A, de Argandoña JD (2008) From diagnosis to prognosis for forecasting air pollution using neural networks: air pollution monitoring in Bilbao. Environ Model Softw 23(5):622–637

India at a Glance (2019) Population enumeration data. https://www.india.gov.in/india-glance/profile. Accessed 9 Dec 2019

Jiang D, Zhang Y, Hu X, Zeng Y, Tan J, Shao D (2004) Progress in developing an ANN model for air pollution index forecast. Atmos Environ 38(40):7055–7064
Kalapanidas E, Avouris N (2001) Short-term air quality prediction using a case-based classifier. Environ Model Softw 16(3):263–272
Ketu S, Agarwal S (2015) Performance enhancement of distributed K-Means clustering for big Data analytics through in-memory computation. In 2015b eighth international conference on contemporary computing (IC3), August 2015b. IEEE, pp 318–324
Ketu S, Mishra PK (2020) Performance analysis of machine learning algorithms for IoT-based human activity recognition. In: Sengodan T, Murugappan M, Misra S (eds) Advances in electrical and computer technologies. Springer, Singapore, pp 579–591
Ketu S, Mishra PK (2021a) A hybrid deep learning model for COVID-19 prediction and current status of clinical trials worldwide. Comput Mater Continua 66(2):1896–1919
Ketu S, Mishra PK (2021b) Internet of healthcare things: a contemporary survey. J Netw Comput Appl 192:103179
Ketu S, Mishra PK (2021c) Cloud, fog and mist computing in IoT: an indication of emerging opportunities. IETE Tech Rev. https://doi.org/10.1080/02564602.2021.1898482
Ketu S, Mishra PK (2021d) Scalable kernel-based SVM classification algorithm on imbalance air quality data for proficient healthcare. Complex Intell Syst 7(5):2597–2615
Ketu S, Mishra PK (2021e) Enhanced Gaussian process regression-based forecasting model for COVID-19 outbreak and significance of IoT for its detection. Appl Intell 51(3):1492–1512
Ketu S, Mishra PK (2022a) Empirical analysis of machine learning algorithms on imbalance electrocardiogram based arrhythmia dataset for heart disease detection. Arab J Sci Eng 47(2):1447–1469
Ketu S, Mishra PK (2022b) India perspective: CNN-LSTM hybrid deep learning model-based COVID-19 prediction and current status of medical resource availability. Soft Comput 26(2):645–664
Ketu S, Mishra PK (2022c) Hybrid classification model for eye state detection using electroencephalogram signals. Cogn Neurodyn 16(1):73–90
Ketu S, Mishra PK (2022d) A contemporary survey on IoT based smart cities: architecture, applications, and open issues. Wirel Person Commun. https://doi.org/10.1007/s11277-022-09658-2
Ketu S, Prasad BR, Agarwal S (2015) Effect of corpus size selection on performance of map-reduce based distributed k-means for big textual data clustering. In: Proceedings of the sixth international conference on computer and communication technology, September 2015a, pp 256–260
Ketu S, Mishra PK, Agarwal S (2020) Performance analysis of distributed computing frameworks for big data analytics: hadoop vs spark. Comput Sist 24(2):669–686
Kurt A, Oktay AB (2010) Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. Expert Syst Appl 37(12):7986–7992
Kyrkilis G, Chaloulakou A, Kassomenos PA (2007) Development of an aggregate Air Quality Index for an urban Mediterranean agglomeration: relation to potential health effects. Environ Int 33(5):670–676
Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22
Lightstone SD, Moshary F, Gross B (2017) Comparing CMAQ forecasts with a neural network forecast model for PM 2.5 in New York. Atmosphere 8(9):161
Man CK, Gibbins JR, Witkamp JG, Zhang J (2005) Coal characterisation for NOx prediction in air-staged combustion of pulverised coals. Fuel 84(17):2190–2195
Mishra M (2019) Poison in the air: Declining air quality in India. Lung India: Official Organ of Indian Chest Society 36(2):160
Nagelkerke NJ (1991) A note on a general definition of the coefficient of determination. Biometrika 78(3):691–692
Ni XY, Huang H, Du WP (2017) Relevance analysis and short-term prediction of PM2.5 concentrations in Beijing based on multi-source data. Atmos Environ 150:146–161
Northey SA, Mudd GM, Werner TT (2018) Unresolved complexity in assessments of mineral resource depletion and availability. Nat Resour Res 27(2):241–255
Packtpub (2018) Machine learning algorithms. https://www.packtpub.com/product/machine-learning-algorithms-second-edition/9781789347999. Accessed 15 May 2022
Pérez P, Trier A, Reyes J (2000) Prediction of concentrations several hours in advance using neural networks in Santiago, Chile. Atmos Environ 34(8):1189–1196
Ruggieri M, Plaia A (2012) An aggregate AQI: comparing different standardizations and introducing a variability index. Sci Total Environ 420:263–272
The World Bank (2019) Population total—India. https://data.worldbank.org/indicator/SP.POP.TOTL?locations=IN. Accessed 9 Dec 2019

Tso GK, Yau KK (2007) Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. Energy 32(9):1761–1768

Vitousek PM (1994) Beyond global warming: ecology and global change. Ecology 75(7):1861–1876

Weisberg S (2005) Applied linear regression, vol 528. Wiley, New York

Yan K, Zhang D (2015) Feature selection and analysis on correlated gas sensor data with recursive feature elimination. Sensors Actuators B Chem 212:353–363

Yilmaz O, Kara BY, Yetis U (2017) Hazardous waste management system design under population and environmental impact considerations. J Environ Manage 203:720–731

Zhang Q, Jiang X, Tong D, Davis SJ, Zhao H, Geng G et al (2017) Transboundary health impacts of transported global air pollution and international trade. Nature 543(7647):705–709

Zhu S, Lian X, Liu H, Hu J, Wang Y, Che J (2017) Daily air quality index forecasting with hybrid models: a case in China. Environ Pollut 231:1232–1244

PLAGIARISM REPORT

Document Information

Analyzed document	batch 10-1.docx (D170045365)
Submitted	2023-06-08 12:02:00
Submitted by	Hemamalini Siranjeevi
Submitter email	hemajeevi@src.sastra.edu
Similarity	0%
Analysis address	hemajeevi.sastra@analysis.urkund.com

Sources included in the report

Entire Document

Air pollution has become a rising and concerning issue at the same time. In many cities across India, air pollution levels are beyond the maximum and if this trend continues, air pollution can lead to many physiological and ecological disasters. The major pollutants SO2,NO2,PM2.5,PM10.But we are going to focus on PM2.5, as this pollutant is very tiny in size that it is almost microscopic.PM2.5 refers to Particulate Matter(which is a combination of solid particles and liquid droplets) whose size is less than 2.5 micro meters.

In this model we predict PM2.5 concentration in the air using Linear Regression based Recursive Feature Elimination with Random Forest Regression(RFERF).Three models namely Decision Tree, Random Forest Regression and Gradient boosting Regression are used and the best one with optimal features and better accuracy is selected. All the models are compared with different metrics and are visualized through graphical representations.

1.1

Introduction: In this paper, the authors present a model that uses machine learning to predict PM2.5 concentration levels in India. Specifically, they use the linear regression-based recursive feature removal and random forest regression model and compare it to other good machine learning models. Air pollution has become a major environmental problem in India, especially in cities like Delhi where population and energy consumption are increasing rapidly. This article highlights the importance of air quality forecasting to warn people of good air quality and reduce the overall impact of poor air quality on human health. Overall, this article presents a comprehensive approach to predicting air quality in India using machine learning and highlights the urgent need to address air pollution. The purpose of this system is to create a model to predict PM2.5 concentration in air less than 2.5 microns using iterative feature removal based on linear regression with random forest regression (RFERF) and gradient boost regression.

Fig 1

1.2 proposed System: The proposed system is a machine learning model for predicting PM2.5 concentration present in the air. It uses a Linear Regression based Recursive Feature Elimination with Random Forest Regression(RFERF) algorithm and also Gradient Boosting Regressor. RFERF basically includes data cleaning, recursive feature elimination ,hyper-tuned Random Forest Regression, and Performance evaluation using different parameters like MSE,MAE,MSLE,RMSE etc. The system chooses the best optimal features and performs the fitting. 1.3 Dataset: This database contains hourly air pollution data from 12 countries that govern air pollution. Air quality data from the Beijing Environmental Monitoring Center. The weather data for each weather station is similar to the closest one from the China Meteorological Administration. The period is from March 1, 2013 to February 28, 2017. Missing data were reported by NA. This dataset which contains 35064 observations is splitted in 80:20 ratio i.e. 80% training data and 20% testing data

Table 1

1.4 Methodology: 1.4.1 Random Forest Regressor:

Random forest regressor is a machine learning algorithm used for regression problems. It is an ensemble