**AIR QUALITY ANALYSIS AND PREDICTIONS IN TAMIL NADU**

**BY:-A.DOMINIC AAKASH**



**INTRODUCTION:-**

Humans can only survive because of air. Its quality must be monitored and understood for our wellbeing. Due to air pollution, millions of people around the2 Journal of Environmental and Public Health being contaminated by hazardous substances. Due to this unchecked pollution, air quality has signifcantly declined.

Data Collection:

1. Gather historical air quality data for Tamil Nadu. You can find this data from government agencies, research organizations, or online platforms like OpenAQ or AQICN.

Data Preprocessing:

2. Clean the data by removing missing values, outliers, and irrelevant columns.

Convert timestamps to datetime objects for time-series analysis.

Perform feature engineering to extract relevant features like temperature, humidity, and wind speed, which can affect air quality.

Exploratory Data Analysis (EDA):

1.Visualize the data to understand patterns and correlations.

Use libraries like Matplotlib and Seaborn for creating plots.

Feature Selection:

2.Identify the most important features using techniques like feature importance or correlation analysis.

Split the Data:Divide the data into training and testing sets for model evaluation.

Machine Learning Model:

1. Choose a suitable machine learning algorithm for air quality prediction. Common choices include Random Forest, Gradient Boosting, or LSTM for time-series data.

2 .Train the model using the training dataset.

COMMUNICATION OF RESULT:

Communicate the insights to the stakeholders using data visualization

3. **Dataset Description and Sample Data**

Te link to the dataset used for this work is given below.

https://www.kaggle.com/rohanrao/air-quality-data-in-

india.

Te dataset includes hourly and daily air quality and AQI

(air quality index) data from numerous stations in several

Indian cities. Te data are for the years 2015 through 2020.

Te original dataset included 29532 rows and 16 columns,

which included all of the cities listed below. Te cities are

given below:

Ahmedabad, Aizawl, Amaravati, Amritsar, Bangalore,

Bhopal, Brajrajnagar, Chandigarh, Chennai, Coimbatore,

Delhi, Ernakulam, Gurugram, Guwahati, Hyderabad, Jaipur,

Jorapokhar, Kochi, Kolkata, Lucknow, Mumbai, Patna,

Shillong, Talcher, Tiruvananthapuram, and

Visakhapatnam.

Te attribute information is given below.

3.1. Date YYYY-MM-DD, City, PM2.5, PM10, NO, NO2, NOx,

NH3, CO, SO2, O3, Benzene, Toluene, AQI, and AQI\_Bucket.

AQI\_Bucket has six values such as good, satisfactory,

dataset is cleanedand moderate, poor, very poor, and severe.

Te selected from the 4 cities datasets such as New Delhi,

Bangalore, Kolkata, and Hyderabad from the original

dataset. Te attribute xylene was removed from the dataset

due to the fact that the column values were empty for all 4

cities chosen by using Microsoft Excel software

The datasetincludes hourly and daily air quality and AQI (air qualityindex) data from numerous stations in 26 Indian cities.

From the original dataset, the data of four cities suchasNeWDelhi, Bangalore, Kolkata, and Hyderabad were extracted.

Because these are major cities of India, it is important to

analyze the pollution levels in diferent urban cities of India

as they are the major contributors to the pollution. Tese

particular cities have a higher population density and give a

good estimate of the pollution.

After cleaning the dataset and dividing it into 4 for each

city, the New Delhi dataset had 176 rows and 15 columns,

the Bangalore dataset had 1362 rows and 15 columns, the

Kolkata dataset had 747 rows and 15 columns, and the

Hyderabad dataset had 1615 rows and 15 columns, re-

spectively. Te sample dataset for New Delhi, Bangalore,

Kolkata, and Hyderabad is shown in Tables 2–5, respectively.

Te initial dataset has an imbalanced composition.

the synthetic minority oversampling technique (SMOTE)

algorithm, the imbalanced dataset is transformed into a

balanced dataset. Oversampling is employed in this algo-

rithm. Any classes with inadequate rows are supplemented

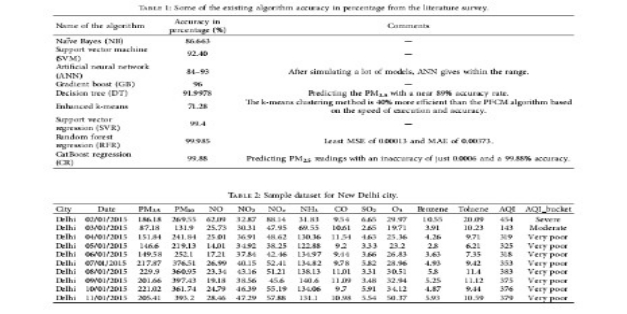
with additional rows to ensure that each class label has an

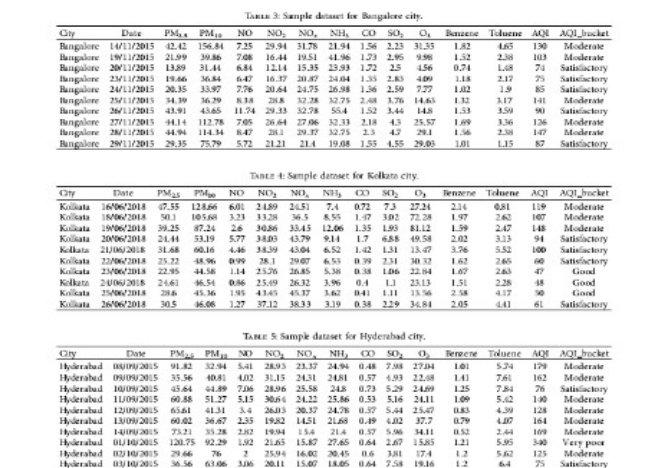
equal number of rows, or more or fewer rows, in the dataset.

Asymmetry exists in an imbalanced dataset. An imbalanced

dataset produces a skewed class distribution, which afects

the model’s accuracy in several ways.





Methodology

In this paper, the proposed methods use three diferent

algorithms to draw a comparative analysis of the AQI values

of New Delhi, Bangalore, Kolkata, and Hyderabad by using

parameters such as PM2.5, PM10, NO, NO2, NOx, NH3, CO,

SO2, O3, Benzene, and toluene levels, which will then

compare the three algorithms and fnd the most accurate and

efcient algorithm.

Te aim is to analyze and present it in an

efcient way. It would help us discover interesting and

insightful information. Tese particular cities have a higher

population density and give a good estimate of the pollution

in a major South Asian city.

More cities have not been added

due to the fact that it makes the research paper way too

lengthy.

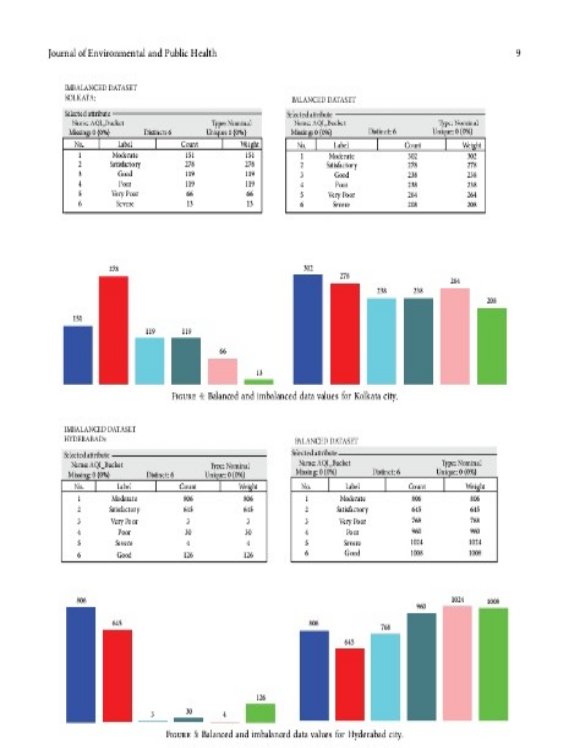
Hence, the major cities of India have been chosen to

analyze the pollution levels in diferent urban cities of India

as they are the major contributors to pollution.

Some of the existing algorithms used are Naive Bayes-a

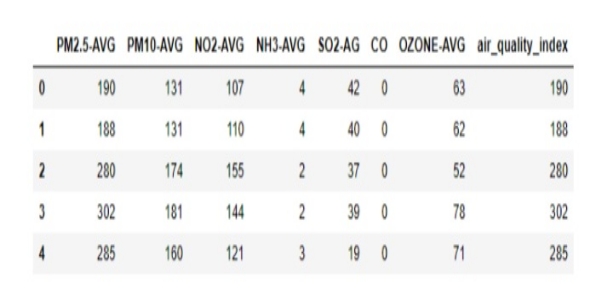
Bayes theorem-based classifer, support vector machine-a supervised learning model for classifcation and regression, artifcial neural network-learning methodology inspired by actual



Program

#importing pandas module for data frame  
  
import pandas as pd  
   
# loading dataset and storing in train variable  
  
train=pd.read\_csv('AQI.csv')  
   
# display top 5 data  
train.head()"

Output



Program:

# importing Randomforest

from sklearn.ensemble import AdaBoostRegressor

from sklearn.ensemble import RandomForestRegressor

# creating model

m1 = RandomForestRegressor()

# separating class label and other attributes

train1 = train.drop(['air\_quality\_index'], axis=1)

target = train['air\_quality\_index']

# Fitting the model

m1.fit(train1, target)

'''RandomForestRegressor(bootstrap=True, ccp\_alpha=0.0, criterion='mse',

                      max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,

                      max\_samples=None, min\_impurity\_decrease=0.0,

                      min\_impurity\_split=None, min\_samples\_leaf=1,

                      min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

                      n\_estimators=100, n\_jobs=None, oob\_score=False,

                      random\_state=None, verbose=0, warm\_start=False)'''

# calculating the score and the score is 97.96360799890066%

m1.score(train1, target) \* 100

# predicting the model with other values (testing the data)

# so AQI is 123.71

m1.predict([[123, 45, 67, 34, 5, 0, 23]])

# Adaboost model

# importing module

# defining model

m2 = AdaBoostRegressor()

# Fitting the model

m2.fit(train1, target)

'''AdaBoostRegressor(base\_estimator=None, learning\_rate=1.0, loss='linear',

                  n\_estimators=50, random\_state=None)'''

# calculating the score and the score is 96.15377360010211%

m2.score(train1, target)\*100

# predicting the model with other values (testing the data)

# so AQI is 94.42105263

m2.predict([[123, 45, 67, 34, 5, 0, 23]])

Output

RandomForestRegressor score: 97.96%

Predicted AQI using RandomForestRegressor: 123.71

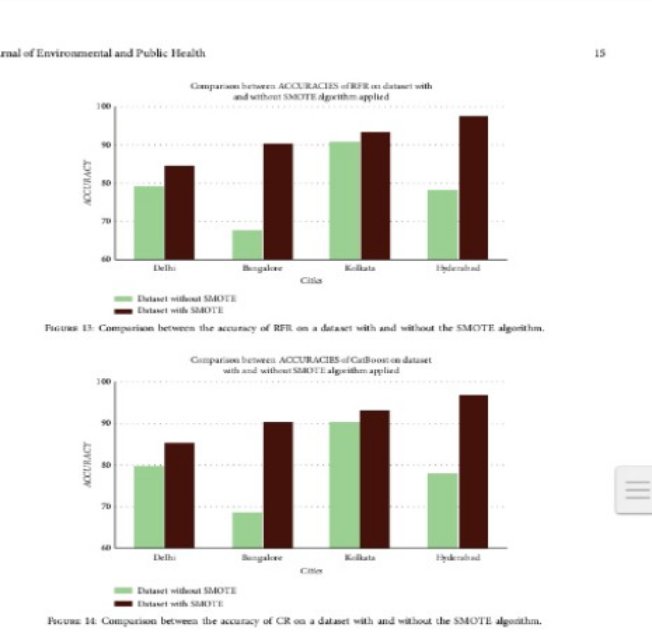
AdaBoostRegressor score: 96.15%

Predicted AQI using AdaBoostRegressor: 94.42

Metrics used:-

Te metrics used in the proposed work are R-SQUARE,mean squared error (MSE), root mean squared error(RMSE), mean absolute error (MAE), and accuracy.

1. R-SQUARE indicates to what extent the regressionmodel is in line with the observed data. A higher Rsquare value denotes a better model ft, the R Squareequation is shown by equation



Conclusion:

This helps the stakeholders to understand about the data insights.