International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(2): 1310-1327 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i02.08199

Application of Mathematical Models for Prediction of Air Quality of Select Indian Cities Through Ensemble Learning Approach

R Vetri Selvi*, R Sathish Babu

Department of Computer & Information Science, Faculty of Science, Annamalai University, Annamalai Nagar, Tamil Nadu, India. *Corresponding Author's Email: vetriselvischolar@gmail.com

Abstract

Clean air is considered an important factor for human and environmental health. Machine learning based prediction models play a prominent role in improving and monitoring the air quality systems and assist in handling different environmental threats. The aim of this research is to develop mathematical models that facilitate ensemble learning for prediction of Air Quality Index (AQI) of select Indian cities. Indian cities studied in this research are Mumbai, Chennai, Bangalore and Delhi. Two models have been developed in this research. The data used for training and testing of the model is accessed from Kaggle database. Air Quality Index is predicted through random forest and gradient boosting models. Performance evaluation of the models has been done by the evaluation metrics RMSE, MAE, MSE and R^2. The gradient boosting model obtained a value of 99.92% (0.9992). Similarly, the random forest model obtained 85.7% (0.8574) as the R^2 score. It has been identified that Gradient boosting model outperforms random forest model by rendering an accuracy of 99%. At the same time random forest model is found to render only 86% accuracy. In addition to that, conversion of the predictions into AQI categories have been evaluated, post AQI regression, through the classification measure F1 and the confusion matrix and the results have been presented for every city in this research.

Keywords: Air Quality Prediction, Ensemble Model, Gradient Boosting, Machine Learning, Random Forest.

Introduction

Clean air is undoubtedly one of the essential factors for the health and survival of both humans and environment. Regular monitoring and prediction of air contamination is important to maintain good air quality. In this context, machine learning has become a potential technique in predicting the Air Quality Index (AQI) (1).

Increasing employment and other resources create opportunities for many industries, however, can cause air pollution (2). Chennai city has numerous industries that create air pollution (3). Major industries are chemical, fertilizer, petroleum and cement plants (4). In Bangalore a substantial change in the concentration of the contaminants has been observed during Covid-19 (5). The time interval from pre-lockdown to unlock period has a substantial role in the difference of air quality (6).

Prediction of AQI index through techniques like XGBoost, k-NN and linear regression have gained potential importance in the recent times (7). AQI prediction model that integrates the AQI spatial patterns has also been studied (8). Machine learning model with a two-stage feature selection has been developed to predict air quality (9). A hybrid machine learning model integrating nonlinear auto regressive average model and deep neural network (DNN) has also been developed (10). Machine learning models have been found to be more economical than traditional techniques (11).

Parameters such as humidity, temperature and wind direction have been used to check the air quality. Some of the advanced models like multilayer perceptron (MLP), SVMs and decision trees (DT) have found to analyse ozone concentration (12). Studies have also employed various advanced deep learning methods for prediction of air quality (13). To predict the air quality index, components as SO2, PM2.5, NO2 and CO are being used (14).

ML algorithms can forecast air quality with greater accuracy (15). However, identifying the sources of pollution and developing practical plans is difficult due to their lack of transparency (16). Integrating

(Received 13th September 2025; Accepted 13th March 2026; Published 21st April 2026)

machine learning with traditional techniques of monitoring air quality can help authorities in identifying the causes and impacts of air pollution to create practical plans in reducing its adverse effects on environment and human health.

Aim of the Research

The aim of this research is to develop an ensemble model-based approach for air quality prediction among selected Indian cities.

AQI Prediction Using Gradient Boosting

Different ensemble regressors were deployed and tested on structured datasets in recent research. It was identified that XGBoost outperformed all the regressor models by capturing difficult patterns in ecological data and giving accurate forecasts (17).

Comparative research was conducted on the performance of the several regression models in predicting the impact of AQI (18). It was found that prediction accuracy of ordinal probit regression and ordinal logit regression model were relatively high.

Training and testing of models based on past and current air quality records containing ecological information such as pollutant concentrations, humidity and temperature revealed that ensemble-based approach with XGBoost and linear regression had better predictive capability (19).

A study on the benefits of algorithms such as LightGBM, XGBoost and ensemble stack of light GB machine and XGBoost showed that ensemble algorithms predicted AQI efficiently (20).

Air quality data of 7 years were collected based on daily intervals in China and evaluated using ensemble models (21). It was identified that stacked model performed better than other models.

Recent research that adopted XGBoost, RF and neural network revealed that neural network needs more tuning parameters and more time (22).

A recent study was conducted to develop a classification algorithm with gradient boosting that rendered scalability and prevented overfitting (23). This research aimed to categorize air quality into three namely unhealthy, moderate and good. It was clear that XGBoost algorithm could be adopted for classification measurements in air quality.

AQI Prediction Using Random Forest

Recent research on the role of supervised methods in ML in the prediction of air quality on based on past and present ecological data revealed random forest was efficient in predicting air quality (24).

Machine learning models such as RF, decision tree, XGB regressor were applied for predicting air quality index (AQI) based on pollutant levels (CO, NO2, PM10, PM2.5). It was found that random forest rendered the highest R^2 score (25).

Research on applying random forest for identifying factors that have high effect NO2 and CO revealed that for carbon monoxide, factors like 24 percent elevation, 33 percent relative humidity (RH), 12 percent wind direction (WD), 10 percent LST (land surface temperature) had high effects (26).

The benefits of adopting Random Forest Regression in identifying difficult correlations between indicators of air quality and input factor was identified through assessment and hard testing (27).

Data on air pollution containing PM10, PM2.5, SO2, NO2, CO, O3 were collected from air pollution monitoring units on airport premises for predicting APSI using Random Forest (28). It was identified that maximizing green spaces could reduce air pollution.

Random Forest classifier was used to determine the air pollution of the surroundings through estimation NO2, PM10 and PM2.5 levels (29). It was found that particulate matter affects human health, government policies and city management.

Research Gap

The prediction of AQI using machine learning models have gained a huge interest int the recent times. Gradient boosting technique has been applied for air quality prediction (17-20). Likewise random forest has been used separately for prediction of air quality (21-27). The present approach implements ensemble learning technique which calculates AQI using CPCB formula and predicts air quality of major Tier-1 Indian cities.

Methodology

Dataset Description

Dataset is collected from the Kaggle database (30). The dataset had values of Air quality Index (AQI) collected daily and on an hourly basis from all over Indian cities via various monitoring stations. The timeframe considered here is for 10 years ranging from 2015 to 2024. Major Indian cities selected for predicting AQI in this research are Chennai,

Mumbai, Bangalore and Delhi. The reason for selecting these four Indian cities among others is because they have been identified most polluted cities that acts as major metropolitans that contribute to political, cultural and economic factors of India (31). These four cities are also known as the top Tier-1 cities, which also have been increasingly identified for its low air quality and heavy air pollution. Air Quality Index is calculated taking into consideration The parameters used for estimating AQI is presented in the Table 1.

Table 1: Parameters Used for AQI Estimation

Parameter	Measured in Unit(s)
PM2.5 (Particulate_Matter 2.5-micrometer)	ug/m3
PM10 (Particulate_Matter 10-micrometer)	ug/m3
NO_2 (Nitric_Dioxide)	ug/m3
NOx (Any Nitric_x-oxide)	ppb: parts-per-billion
NH3 (Ammonia)	ug/m3
CO (Carbon_Monoxide)	mg/3
SO_2 (Sulphur_Dioxide)	ug/m3
O_3 (Tri-Oxygen or Ozone)	ug/m3

Algorithms

Random Forest (RF) Algorithm:

RF is a famous algorithm in machine learning that falls under supervised learning. It was adopted for regression and classification issues in machine learning. It employs ensemble learning, which integrates several classifiers for solving complicated issue and to enhance the model performance. RF is a classifier which builds multiple decision trees on random samples of the dataset and merges their results, mostly using ensemble technique like maximum voting or averaging to enhance the predictive accuracy and prevent overfitting. Rather depending on single decision tree, RF gathers predictions from each decision tress and predicts the result based on the majority voting mechanism. Increasing the number of trees improves the model accuracy and minimizes the issue of overfitting. RF integrates ensemble of trees to classify the dataset; it is likely that certain decision trees might yield the right result and some may not. Overall, each tree accurately classifies the input data. Thus, the two criteria below are important to enhance the stronger random forest model. Exact classification relies on the valid feature values instead of arbitrary or empty values Results of each tree must be least correlated (32).

RF algorithms are used because they take minimal time for training when compared with other algorithms; predicts results with greatest precision particularly for huge datasets it executes effectively; it remains accurate even with huge portion of data is missing.

How RF algorithm works is explained as follows:

This algorithm is done in two phases. Random forest is created first by integrating N decision tree and then estimations made for each tree constructed in previous phase.

Step 1: Data points are selected randomly from the examples for training data

Step 2: Decision trees are built along with chosen subsets or data points

Step 3: Select the number N representing how many trees to be built in the forest

Step 4: Reiterate step 1 and step 2 processes.

Step 5: Estimate the results of new inputs using all decision tree and categorize them based on the class which gets the most vote.

This algorithm would be clear with the following example. For example, a dataset that contains various types of fruit images. RF classifier is adopted here and applied on the dataset. It is categorized into subsets and assigned to each tree in the forest. While training, each tree predicts its own prediction and when new data comes, output would be selected based on the highest number of predictions and RF classifier produces the outcome.

Mathematical Representation

To predict the random forest classification of the air quality index datasets, the following Equation [1] has been adapted:

$$\hat{a} - mode(IT_1(b), IT_2(b), \dots, IT_n(b)) \quad [1]$$

Where, the $IT_1, IT_{2,\dots,} IT_n$ represent individual-trees of forest and the "b" represent the input data-point.

Similarly, to predict the random forest regression, the following Equation [2] is utilized:

$$\hat{a} = \frac{1}{m}\sum_{k=1}^{m} * IT_k(b) \quad [2]$$

Thus, in random forest, the most common prediction is obtained from the mode (text mode) in classification and average predictions is obtained from the regression from estimation 'Σ'.

Gradient Boosting (GB) Algorithm:

GB is an ensemble method which integrates weak classifiers into strong ones. Weak classifiers generate models with weak prediction power; similarly, strong classifiers generate models with strong prediction power. Gradient boosting is powerful and sequential algorithm. In gradient boosting, weights of training samples remain unchanged, without considering classification accuracy. Weight is not assigned to each sequential model in gradient boosting. Differentiable loss function is optimized of a weak classifier through sequential boosting steps. The result of gradient boosting is preservative models of many weak classifiers. Decision trees are used in gradient boosting and at the same time, it doesn't use stumps in decision tree. The first model in this algorithm is root node in decision tree (where classification is done by voting and regression uses averaging). Subsequently gradient boosting models are deeper ones. The depth is mostly identified by the researcher. Values like 32 or 8 are common, based on the difficulty of task or dataset. Depth of the tree is a hyper parameter needs to be adjusted (33).

How gradient boosting works are explained below:

Step 1- Configuration: It adopts training dataset to build a base learner, using a decision tree. Here, predictions are done randomly during initial stages. The decision tree has only new end nodes. Mostly they are selected based on transparency, these base or weak learners act as strategic entry point. Such setup builds the path for next iterations to be refined.

Step 2- Computing residuals: For every training, residual error by calculating by identifying the difference between actual and predicted value. Thus, step checks places where the predictions of the model require enhancement

Step 3-Improving through regularization: After computing the residual and continues to train new model, regularization process happens. This stage cut down the impact of every new learner which is weak is combined with ensemble. This helps to prevent overfitting and overall optimization performance.

Step 4- Training new model: Residual errors determined in before step is used and weak learner or new model is trained for predicting them exactly. This step corrects the errors made from existing models and enhances the overall result.

Step 5- Updating the combined model: Performance of newly trained and combined model is computed by adopting independent test data. When estimation on the independent test data is sufficient, ensemble integrates new learner; otherwise, hyper parameters need to be tuned.

Step 6- Replication: Reiterate the above steps as needed. Successive iterations enhance and improve the base model by incorporating new trees which resulted in the best overall performance. If the update ensemble and last model meet expectations accuracy relative to baseline, then proceed to next step.

Step 7- Cut-off criteria: Process of boosting is ended when it meets its criteria like accuracy threshold, accuracy goal or marginal improvements. These steps ensure that the final output meets expectations of performance between effectiveness and complexity.

Mathematical Representation

The gradient boosting is estimated by using the residual (r_n) from the data point. By utilizing the Equation [3], the gradient boosting model R(x) is mathematically represented:

$$r_n = a_n - R(x_n) \quad [3]$$

$$R(x) \leftarrow R(x) + \eta \quad [4]$$

In ensemble models, to minimize the data point residuals, new tree is trained by utilizing the Equation [4] for every individual tree (iteration). This in turn updates the gradient boosting model with learning rate (lr) to control the tree's impact (i.e.: η).

Research Flow

The machine learning models developed here are trained and tested using two types of algorithms, namely the Random Forest and the Gradient Boosting. The hyperparameters of the models applied have been presented in Table 2 as follows:

Table 2: Hyperparameters of the Models

Hyperparameter	Random Forest	Gradient Boost
Number of estimators	200	300
Loss criterion	'squared_error'	'squared_error'
Max_Depth	NA	4
Subsample	NA	0.1
Learning rate	NA	0.05
min_samples_split	2	2
min_samples_leaf	1	1
Random state	42	42
validation_fraction	NA	0.1
early_stopping	NA	Disabled

As the above Table 3 illustrates, for the random forest model the key parameters are estimators = 200; random state = 42; criterion = 'squared error'; max_depth = None; max_features = 'auto'; bootstrap = True; min_samples_split = 2 and min_samples_leaf = 1

The hyperparameters of Gradient Boosting Regressor are n_estimators=300; learning rate=0.05; max_depth=4; random state=42; subsample=1.0; min_samples_split=2; min_samples_leaf=1 → loss='squared error'; validation fraction=0.1 and early stopping is disabled.

The research design here adopts the data split of 80:20 for training-and-testing, respectively. Training of the ML models is carried out to estimate the AQI using the formulae adopted here. Commonly, the most novice mistake an investigator does when developing research lies with adopting the appropriate evaluation technique. Though there are several techniques available like classification accuracy, log loss, AUC (area under the curve), F1-score, precision, accuracy, recall and more, based on the model developed the measures should be adopted. To measure the developed model's performance regressor based statistical measures like R^2, RMSE (Root-Mean-Squared-Error) and MAE (Mean-Absolute-Error) are adopted, respectively. The obtained values are then compared here, to identify the best model developed.

Similarly, once the AQIs are calculated using the two models, they are classified into respective classifications of the air quality based on six classes (Good, Satisfactory, Moderate, Poor, Very Poor and Severe) based on their ranking obtained from the air quality index level. The classes are categorized into following six AQI bucket as illustrated in Table 3.

Table 3: Classification of AQI Scores

Condition	AQI Bucket Scale
N < 50	Good
N < 100	Satisfactory
N < 200	Moderate
N < 300	Poor
N < 400	Very Poor
N > 400	Severe

The flow diagram of the research is explained in Figure 1. It starts with the data collection, ML model development, data cleansing, training and testing the models, evaluating the models' performance using RMSE, MAE and R^2 scores.

Novelty of the Approach

The approach used is the ensemble approach with random forest and gradient boosting models for AQI predictions. The ensemble model uses two approaches bagging and boosting, where individual models are combined into one superior

model. The same is applied in this study where the ensemble model includes less bias (more flexible) and lesser data-sensitivity (lesser variance) to produce more accurate AQI prediction model. The RF equips the bagging-approach (models are trained parallelly with data subsets) while the GB equips the boosting approach (models are trained sequentially, with previous mistakes). For fine tuning of the model, K-Fold cross-validation is utilized. Individual random forest model gathers the data, splits it into subsets in step 1. Later in step 2, the subsets are transformed as decision trees, where the features are extracted and extracted. In step 3, the model is tested and voted by using the subsets where the processed data is passed to the next step. In step 4, based on the majority vote the best candidate is selected. Figure 2 illustrates the bagging and boosting process.

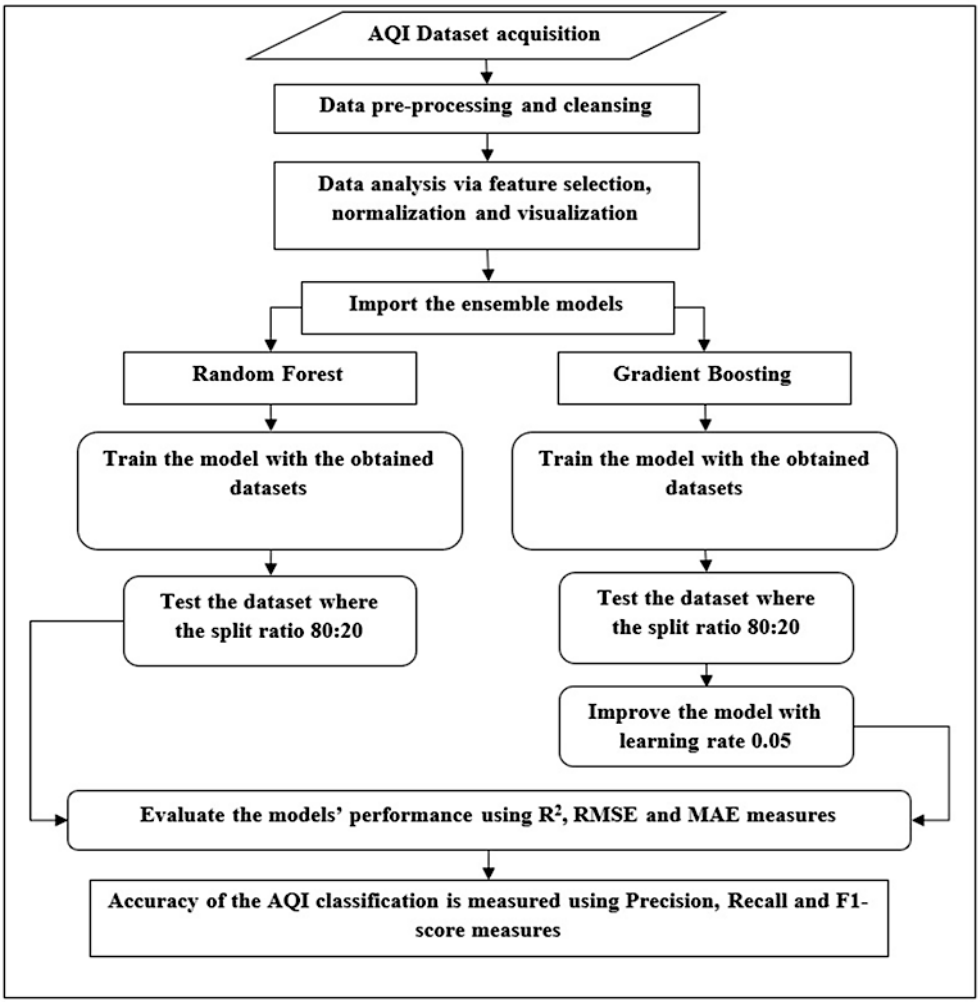

Figure 1: Flow Diagram of the Research

The RF model parallelly works with decision tress and based on the majority voting system, the output is acquired. This output is used as input for the GB model, where it works sequentially with different trees that corrects the mistakes using previous errors as the learning method. Data is prepared, ensemble models are used, K-fold cross-validation is applied for robustness with k-fold set with values as shuffle "true", random state as 42 and split as 5. Once the output is processed the confusion matrix is created in python. Through this approach the air quality index prediction is made for four cities Indian cities namely Delhi, Mumbai, Bangalore and Chennai. Figure 3 illustrates the ensemble learning process in detail.

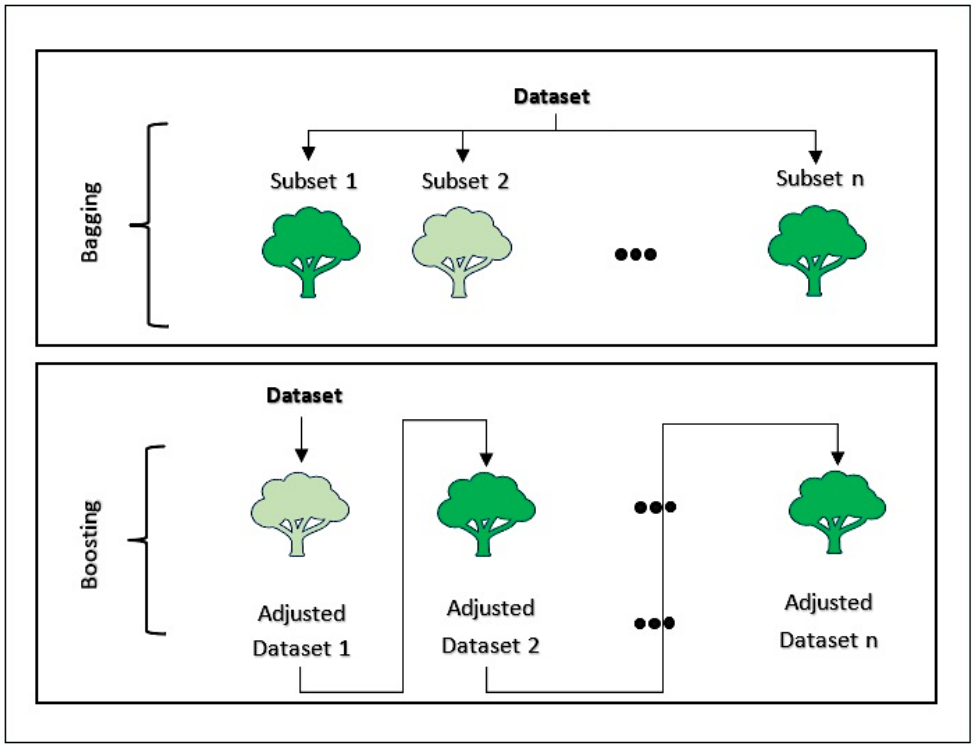

Figure 2: Bagging and Boosting Approach

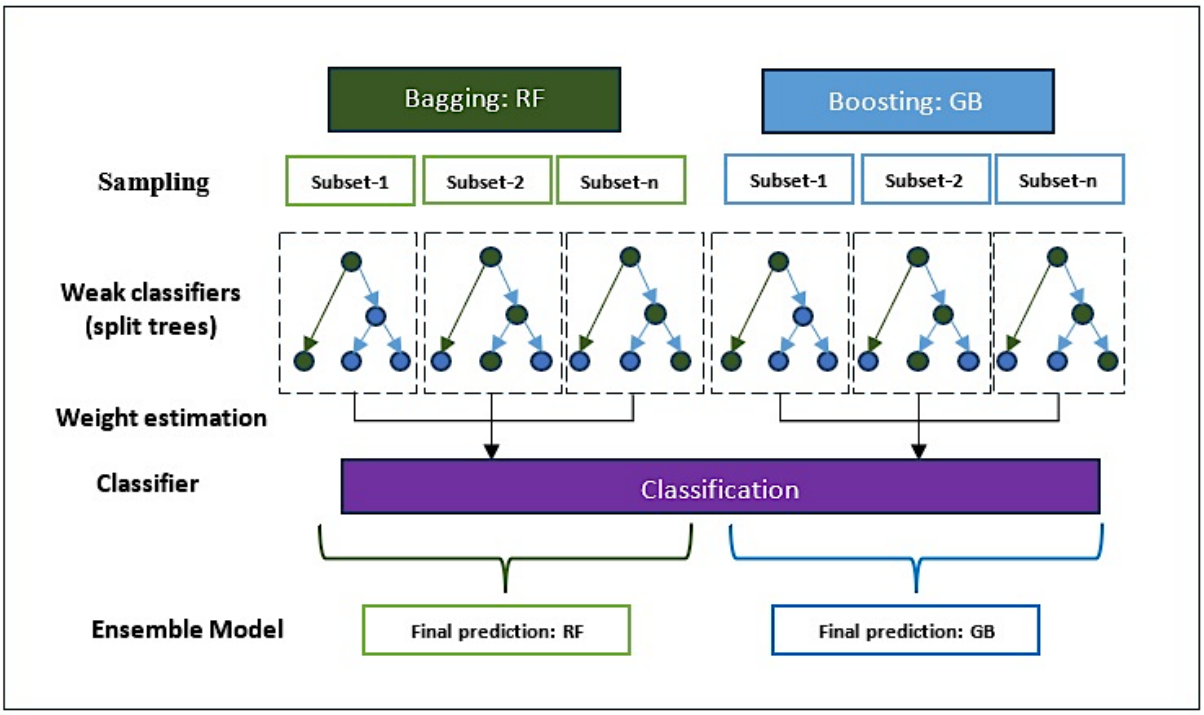

Figure 3: Ensemble Approach for Air Quality Index Prediction

System Specifications

The CPU (central processing unit) used for the AQI prediction is Intel Core-i9 processor. The memory (random memory access: RAM) used for the proposed study is 8GB. The software used for development of the model is Python. The coefficient of determination, i.e., R-squared for a chosen model, gradient boosting on a training dataset points to how well the predictions fit the actual training data. It can be suggested that a higher R-squared value, which is closer to 1, suggests a better fit, implying that the model

explains a greater proportion of the variance in the chosen variable using the features provided.

The main Python libraries considered for the study are Pandas, Numpy, Tensorflow, sklearn and seaborn. The two most popular Panda tools considered for the data analysis are Anaconda and Jupyter Notebook. The flexibility of Python and high-powered libraries make it an outstanding and effective fit when it comes to data analysis experiments, from simple statistical techniques to advanced machine learning algorithms and big data processing.

Evaluation Metrics

The evaluation metrics considered for the study are R^2, MSE, RMSE and MAE. One of the python libraries, i.e., scikit-learn impart implementations for most of these metrics.

R^2 Score

R^2 value refers to the ratio of variance in the air quality index explained by the proposed model. Best possible score for the metric is 1.0 and at times it could be negative. The performance of this metric specifies how effectively the values match with actual values predicted has been presented in Equation [5].

$$R^2 = \mathbf{1} - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \quad [5]$$

Here, y_i refers to predicted data, whereas y_i refers to actual data and $\underline{y_i}$ signifies the mean value.

MSE (Mean-Squared-Error) And Rooted-mean Square Error (RMSE):

MSE represents the actual and predicted values' average squared differences. RMSE represents nothing but the square root of the average of the squared difference between the obtained value and the actual value calculated by the model. It is given in the Equation [6]. It is square root of MSE, which is known as mean square error. The mathematical representation is given in the Equation [7].

$$MSE = \frac{1}{M} \times \sum_{x=1}^{M}(y_i - \hat{y}_i)^2 \quad [6]$$

$$RMSE = \sqrt{MSE} \quad [7]$$

$$MAE = \frac{1}{M} \times \sum_{x=1}^{M}|y_i - \hat{y}_i| \quad [8]$$

$$F1\ Score = 2\ X \frac{Recall\ X\ Precision}{Precision + Recall} \quad [9]$$

Mean Absolute Error (MAE):

MAE is generally calculated based on the value of how close a measured line is to data points. It is also denoted as measure of inaccuracies between paired observations stating similar phenomenon. It is presented in the Equation [8]. The equation of F1 Score is presented in the Equation [9].

AQI Estimation:

The AQI formula to estimate the air quality for the four cities uses the parameters PM2.5, PM10, SO_2, NOx, NH_3, CO and O_3, from the dataset where the following conditions must be met:

At-least one of the indices 'PM2.5 or PM10' must be included with at-least three sub-indices (SO_2, NOx, NH_3, CO and O_3).

The average hourly values of the indices PM2.5, SO_2, NOx, PM10 and NH_3 has 24-hours as standard values with at-least 16 values present in it.

The CO and O_3 have 8-hours as standard value with 16 minimum values in it.

AQI is predicted based on the procedure suggest by CPCB (Central Pollution Control Board) of the Indian Government. The ensemble model works on pre-defined AQI buckets with which the index levels are estimated. Though there are no upper or threshold value of AQI, it's very rare to witness AQI > 1000. The process of prediction of AQI is given in Figure 4.

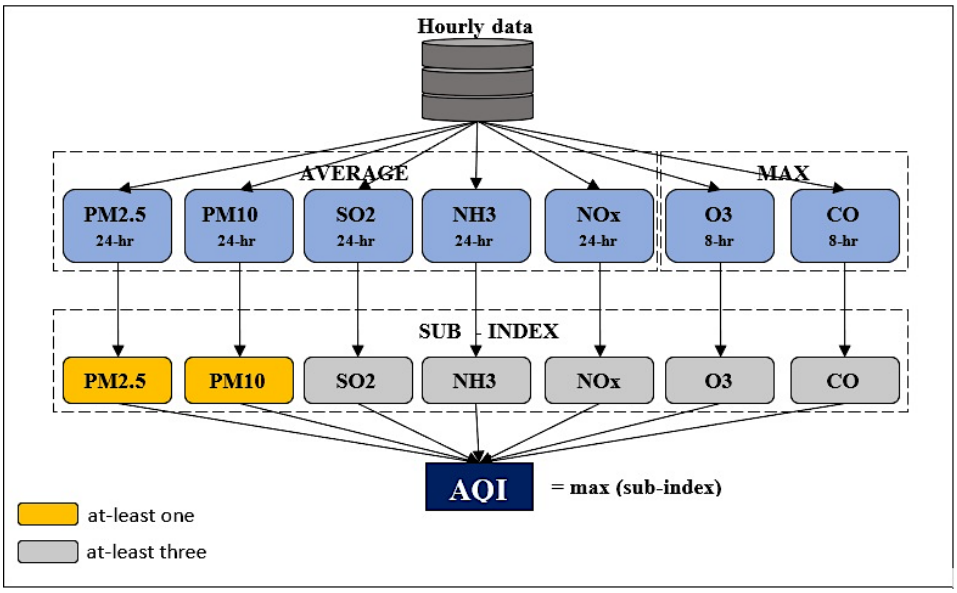

Figure 4: AQI Formula estimation

Results

The study considers a *heat-map*, which is an array wherein the columns refer to one chosen attribute and the rows refers to another. The value of all these cells shows the relationship between the given attributes. For easy evaluation, the values are later converted to a colour map. The results and comparison of gradient boosting and random forest are presented in the current section. To predict air quality, data between 2015 and 2024 from all the major cities, i.e., Bangalore, Chennai, Delhi and Mumbai, have been collected. The results and graphs of air quality indicator are as follows.

AQI prediction using Random Forest

Figures 5(A)-8(A) represent the results of classification metrics using random forest for the cities Bengaluru, Chennai, Delhi and Mumbai respectively. Similarly, the Figures 5(B)-8(B) represent the results of the AQI prediction using random forest for the cities Bengaluru, Chenni, Delhi and Mumbai respectively. The findings can be interpreted as follows.

With regards to **Bangalore** AQI, the obtained value of MAE is 46.76363382772391; MSE is 2978.9684354437254; whereas RMSE value is 54.579927037728126.

With regards to the city **Chennai** the values of MAE, MSE and RMSE of Chennai city are as follows: 46.46988953223046, 2940.5460615597403 and 54.226802059127.

With regards to **Delhi**, components like PM2.5', 'NO2', 'CO', 'SO2', 'O3', have been considered for the study to analyse and monitor the air quality. With regards to Delhi, the obtained value of MAE is 46. 6963096406161, MSE is 2974. 290174889561 and RMSE is 54.53705322887881.

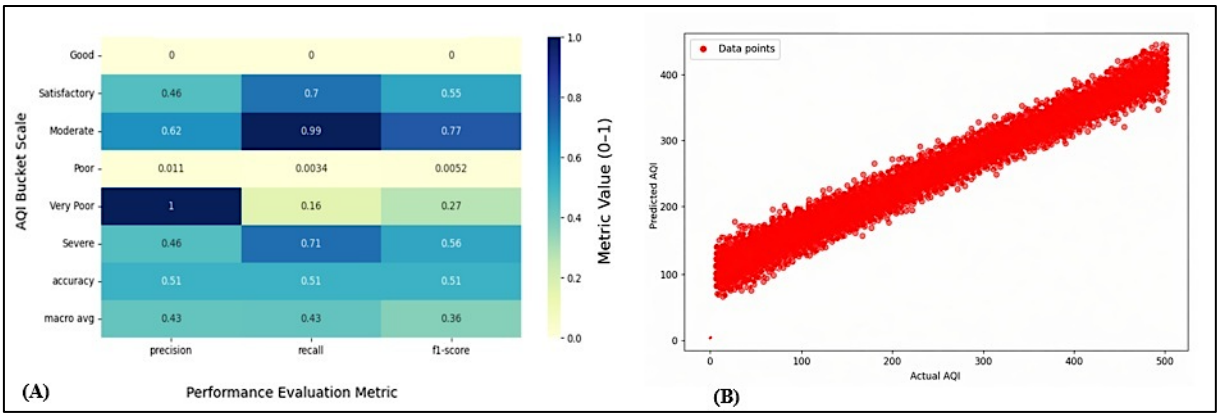

Figure 5: AQI using Random Forest Approach: (A)Bengaluru Classification Metrics Heatmap (per class), (B) Bengaluru Actual vs Predicted AQI

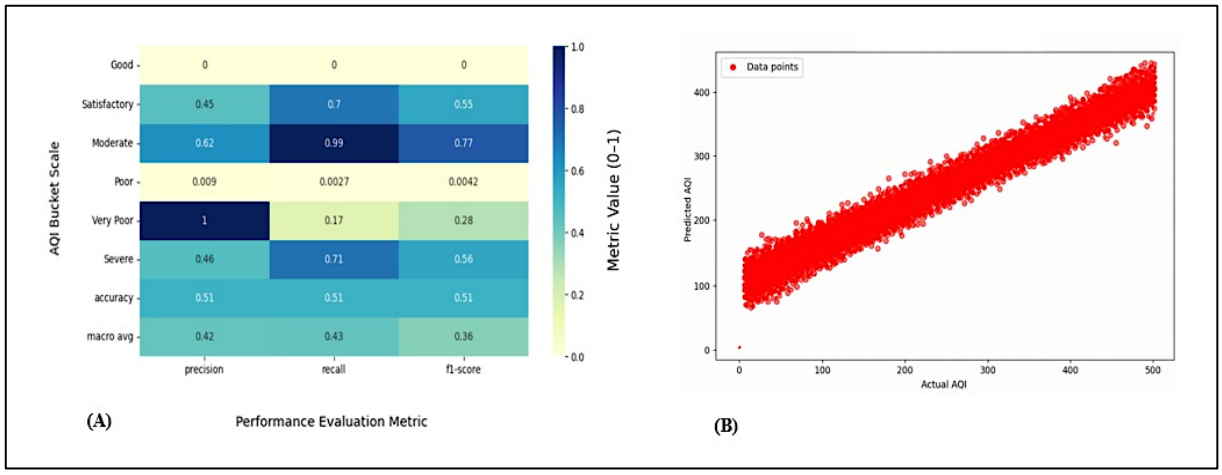

Figure 6: AQI using Random Forest Approach: (A)Chennai Classification Metrics Heatmap (per class), (B) Chennai Actual vs Predicted AQI

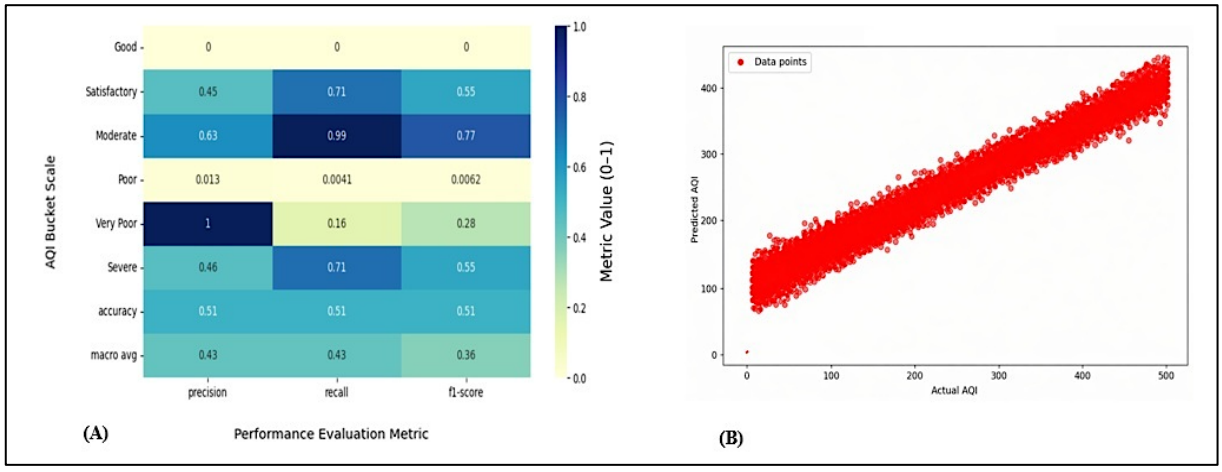

Figure 7: AQI using Random Forest Approach: (A)Mumbai Classification Metrics Heatmap (per class), (B) Mumbai Actual vs Predicted AQI

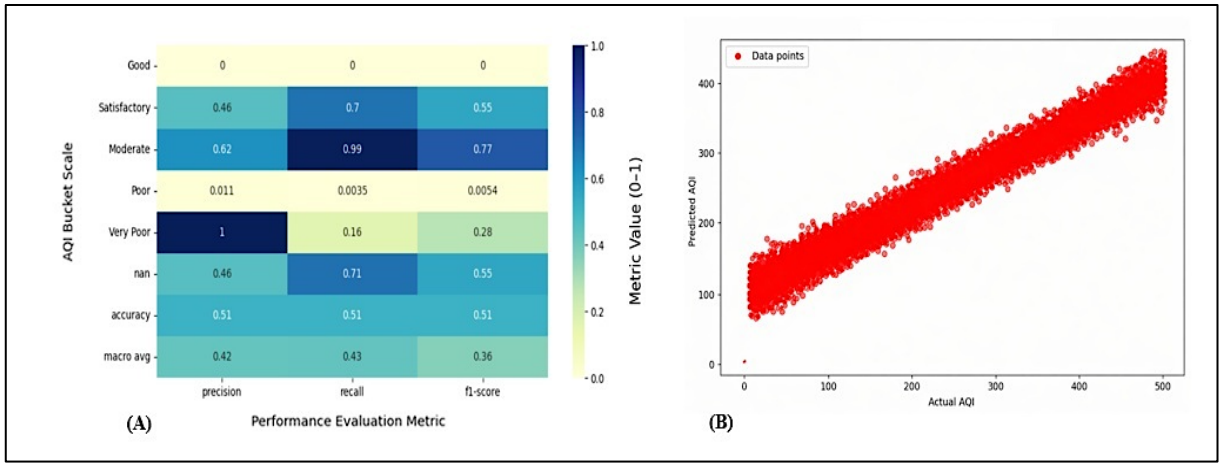

Figure 8: AQI using Random Forest Approach: (A)Delhi Classification Metrics Heatmap (per class), (B) Delhi Actual vs Predicted AQI

The MAE, MSE and RMSE value of **Mumbai** is 46.827017284654886, 2982.570879575442 and 54.61291861433009 respectively.

The following Table 4 summarizes the RMSE and MAE scores of the four cities via random forest model.

Table 4: Performance Evaluation Metrics for Random Forest

Cities	MAE	RMSE	MSE
Delhi	46. 6963096406161	54.53705322887881	2974.2902
Bangalore	46.76363382772391	54.579927037728126	2978.9684
Mumbai	46.827017284654886	54.61291861433009	2982.5709
Chennai	46.46988953223046	54.226802059127	2940.5461

AQI Using Gradient Boosting

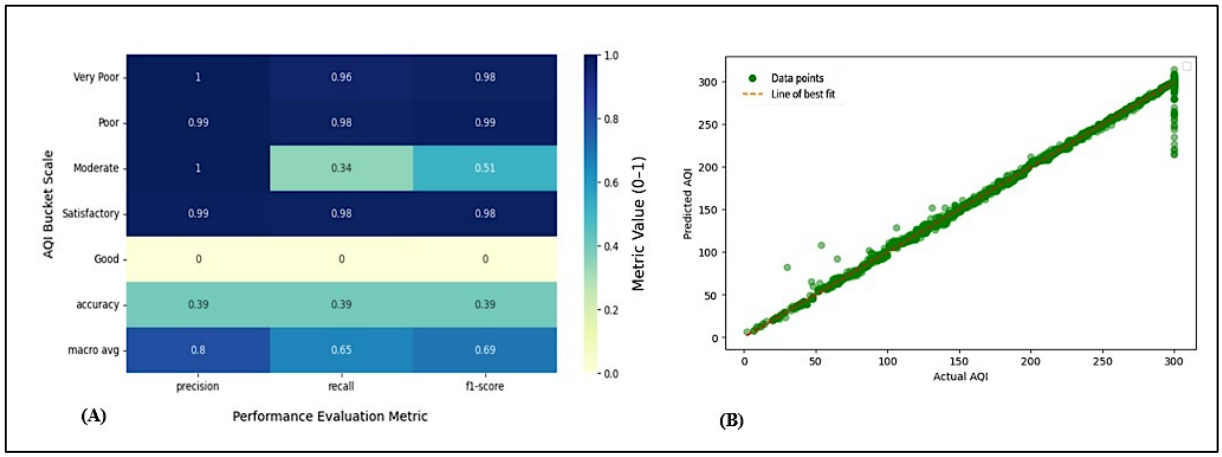

Figure 9: AQI using Gradient Boost Approach: (A)Bengaluru Classification Metrics Heatmap (per class), (B) Bengaluru Actual vs Predicted AQI

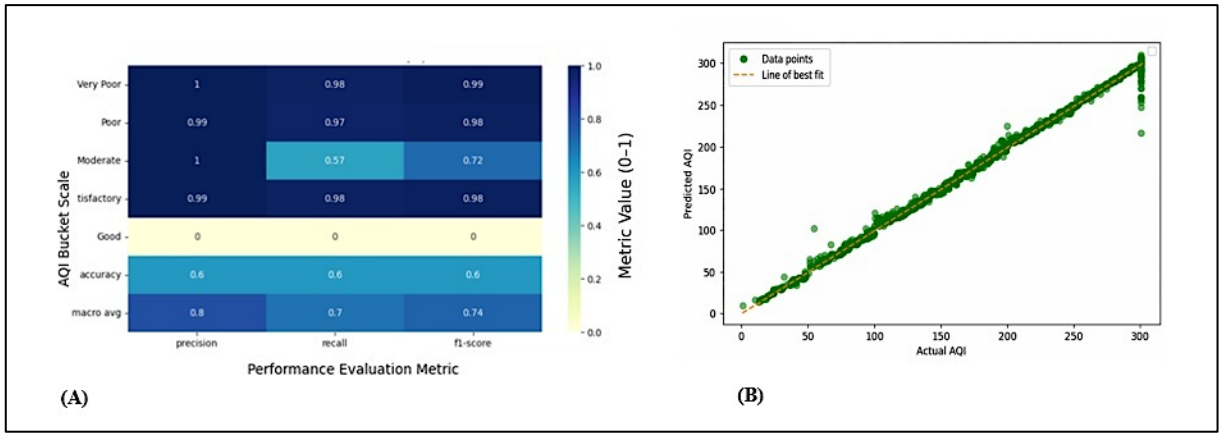

Figure 10: AQI using Gradient Boost Approach: (A)Chennai Classification Metrics Heatmap (per class), (B) Chennai Actual vs Predicted AQI

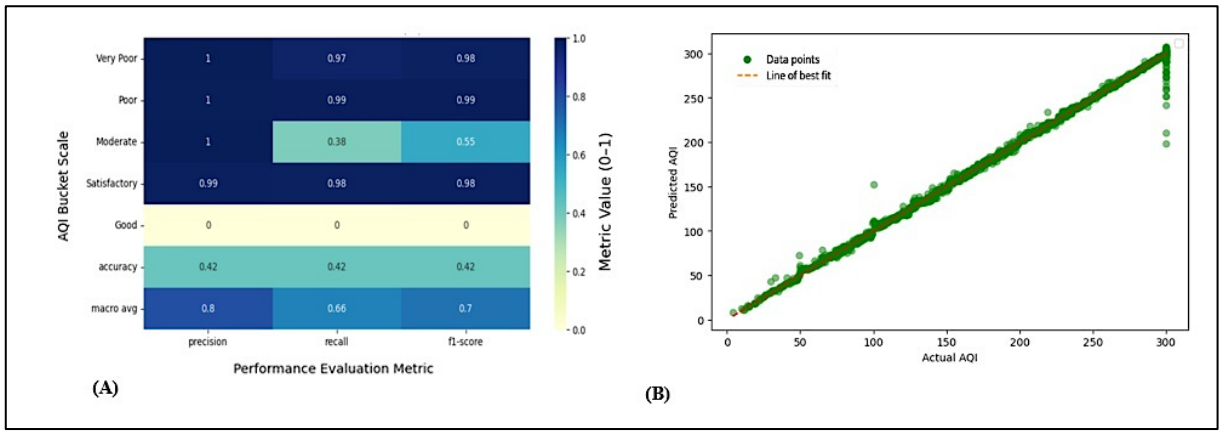

Figure 11: AQI using Gradient Boost Approach: (A)Mumbai Classification Metrics Heatmap (per class), (B) Mumbai Actual vs Predicted AQI

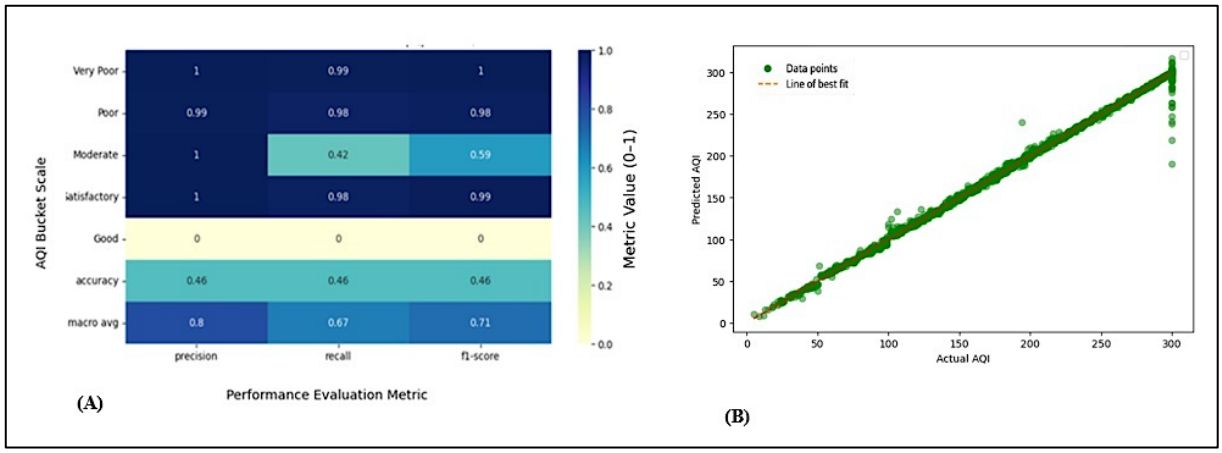

Figure 12: AQI using Gradient Boost Approach: (A)Delhi Classification Metrics Heatmap (per class), (B) Delhi Actual vs Predicted AQI

From the above random forest's performance table, it is evidently proven that there exists huge error rate in the predictions of the model. Henceforth with improvisation (using fine tuning method: learning rate), the same datasets are used in the next model.

Figures 9(A)-12(A) represent the results of classification metrics using random forest for the cities Bengaluru, Chenni, Delhi and Mumbai respectively. Similarly, the Figures 9(B)-12(B) represent the results of the AQI prediction using random forest for the cities Bengaluru, Chennai, Delhi and Mumbai respectively. The findings can be interpreted as follows

The lower values of both RMSE and MAE suggest better accuracy level, whereas the R^2 value of "1" implies better precision and lower MSE value implies that the proposed model is a better fit. Accordingly, the study elaborates on these values in the result section.

The values of **Bangalore** are as follows: RMSE, 2.29, MAE: 0.37, R^2 Score: 0.99.

The scores of **Chennai** show that RMSE value is 1.69, MAE is 0.35 and R^2 Score is 0.99.

The values obtained for **Delhi** are as follows: RMSE: 1.79, MAE: 0.32 and R^2 Score: 0.99.

The scores of **Mumbai**, the RMSE value is 1.91, MAE: 0.33 and R^2 Score: 0.99.

The following Table 5 summarizes the RMSE and MAE scores of the four cities via gradient boosting model. The values prove that the improvised gradient boosting model acquired a better fit than the random forest model. However, the MSE values shows that the model is an average fit and might need more fine tuning of the parameter, which will be carried out in the next upcoming research.

Table 5: Performance Evaluation Metrics for Gradient Boosting

Cities	MAE	RMSE	MSE
Delhi	0.32	1.79	3.2041
Bangalore	0.37	2.29	5.2441
Mumbai	0.33	1.91	3.6481
Chennai	0.35	1.69	2.8561

Post AQI Regression Results- confusion Matrix With K- fold Cross Validation

K-fold cross validation approach was carried out after obtaining AQI value regression result. The K-Fold in ensemble learning is divided into "K" equally sized folds. For every k fold: the model is trained on K-1folds and the remaining folds are used for validation. The fold accuracy is estimated using the mathematical equation [10].

Similarly, the accuracy of mean cross-validation is estimated using the equation [11].

$$\text{ACC} = \frac{\text{Total Correct predictions}}{\text{Total Samples of k Fold}} \quad [10]$$

$$\text{Mean ACC (MA)} = \frac{1}{\text{F}}\sum_{\text{f=1}}^{\text{F}} \text{ACC} \quad [11]$$

Where, F denotes the number-of-folds and ACC represents the fold accuracy estimated.

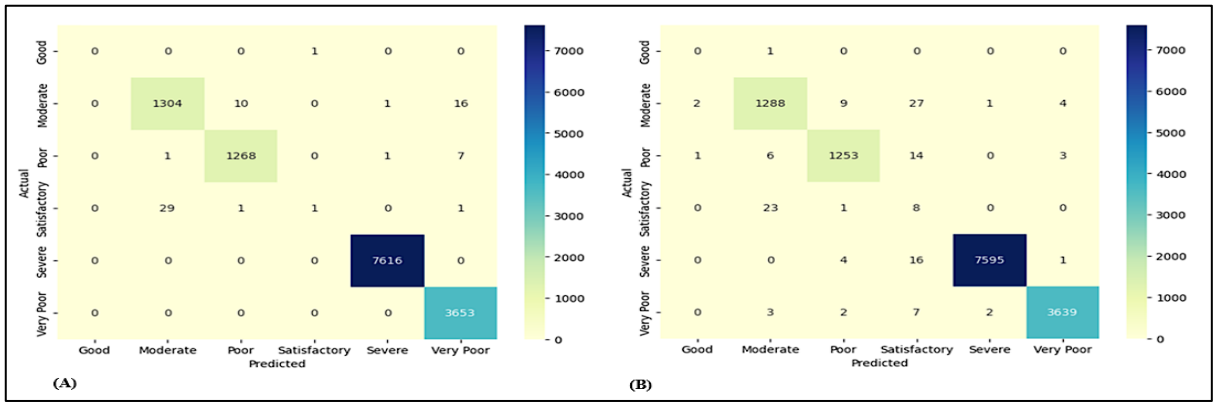

Figure 13: Confusion Matrix Aggregated for K-folds for Chennai: (A) Random Forest, (B) Gradient Boost

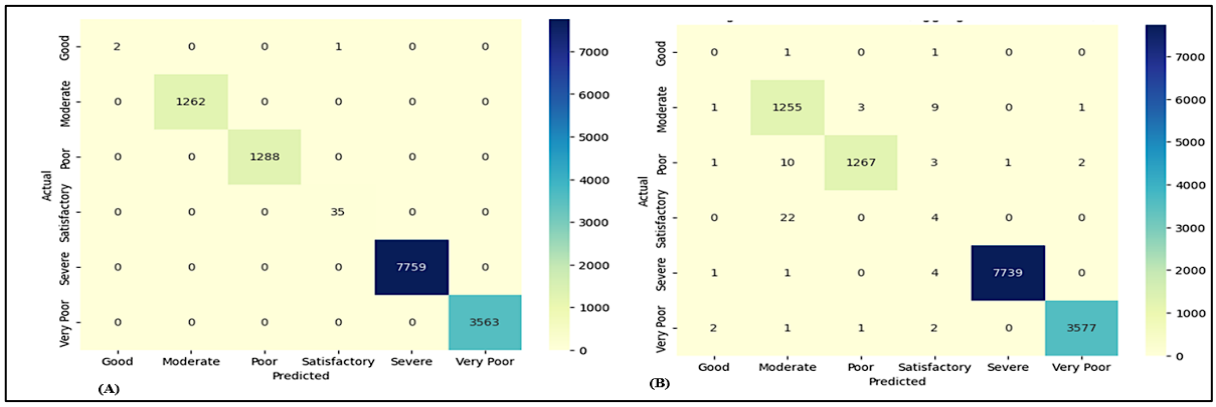

Figure 14: Confusion Matrix Aggregated for K-folds for Bengaluru: (A) Random Forest, (B) Gradient Boost

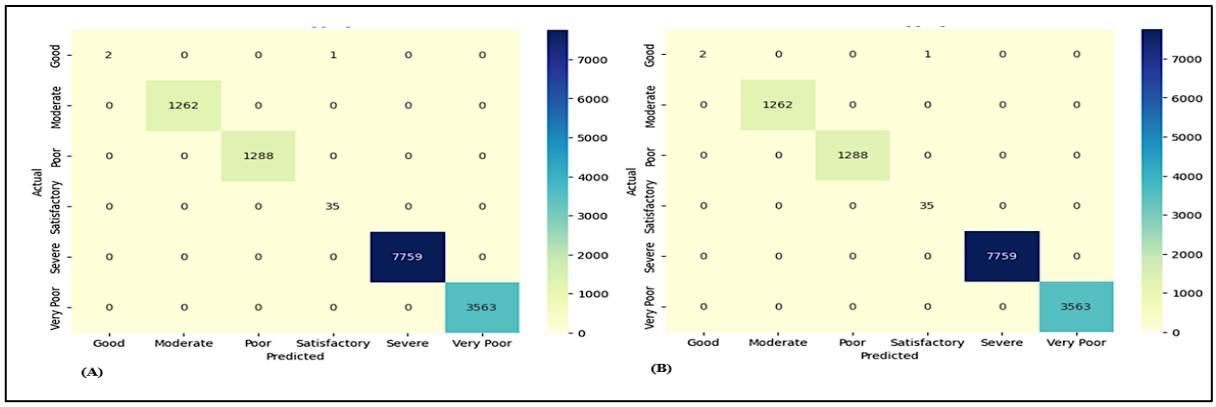

Figure 15: Confusion Matrix Aggregated for K-folds for Mumbai: (A) Random Forest, (B) Gradient Boost

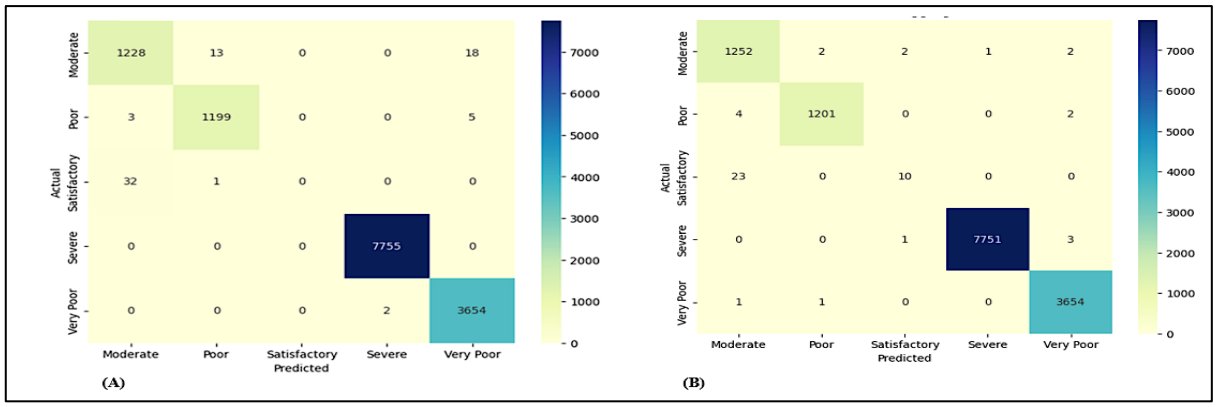

Figure 16: Confusion Matrix Aggregated for K-folds for Delhi: (A) Random Forest, (B) Gradient Boost

The results have been graphically presented in the form of confusion matrix for each of the city that has been considered in the research.

Figures 13(A)-16(A) represent the confusion matrix aggregated over K-folds for the cities Chennai, Bengaluru, Mumbai and Delhi respectively while using the Random Forest model. Similarly, Figures 13(B)- 16(B) represent the confusion matrix aggregated over K-folds for the cities Chennai, Bengaluru, Mumbai and Delhi respectively while using the Gradient Boost model.

Performance evaluation

The performances of the random forest and gradient boosting models are compared for better understanding in this section and presented in Table 6 and Figure 17 as follows.

Table 6: Performance Evaluation Scores of the Models Developed

Cities	MAE	RMSE	MSE
Delhi	0.32	1.79	3.2041
	46. 6963096406161	54.53705323	2974.290175
Bangalore	0.37	2.29	5.2441
	46.76363383	54.57992704	2978.968435
Mumbai	0.33	1.91	3.6481
	46.82701728	54.61291861	2982.57088
Chennai	0.35	1.69	2.8561
	46.46988953	54.22680206	2940.546062

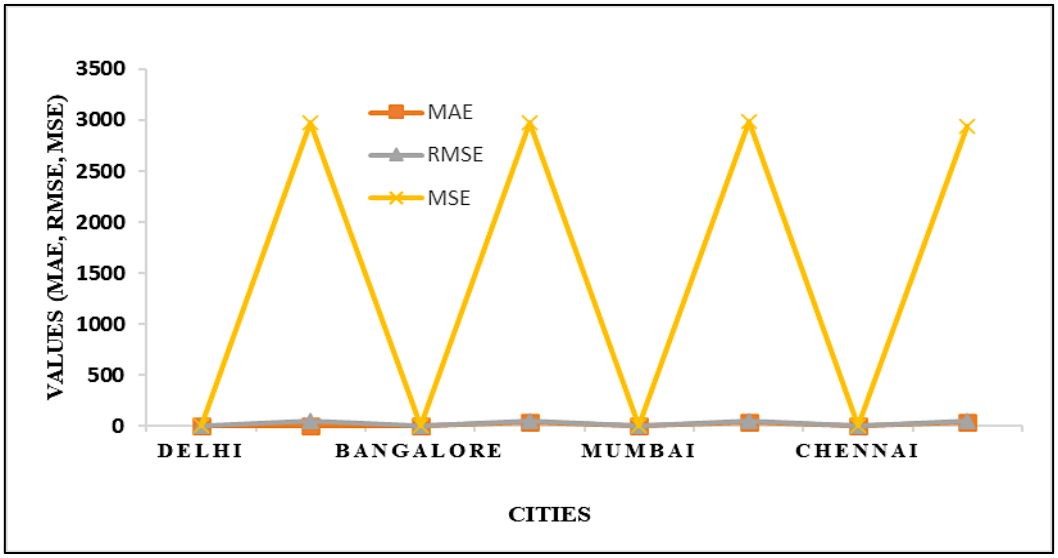

Figure 17: Performance Evaluation of Both Models Developed

From the performance evaluation illustrated in the Figure 17 it is proven that improvised model (gradient boosting) obtained better predictions than the random forest model. The MSE values of the gradient boosting model rapidly improved showing that, the improved gradient boosting model is a better fit.

The results of the performance evaluation of the Random Forest and Gradient Boost model post AQI regression using F1 Score for all the four cities is presented in Table 7.

Table 7: Post AQI Regression Metrics Comparison

Cities	Algorithm	Precision (in %)	Recall (in %)	F1-score (in %)
Chennai	RF	81	78	78
	GB	75	75	75
	RF	77	77	77
Bangalore	GB	70	69	69
	RF	88	85	86
Mumbai	GB	88	85	86
	RF	79	79	79
Delhi	GB	96	86	88

The F1 values show that Mumbai based predictions are the most accurate followed by Delhi, Chennai and Bengaluru.

Discussion

Different machine learning techniques such as random forest, linear regression, Adaboost models, XGBoost, ridge and lasso and KNN for predicting PM2.5 in polluted cities have been adapted in the past (34). AQI prediction using random forest were estimated using the metrics RMSE, MAPE and MAE [54.59, 1.94 and 39.84].

The importance of machine learning methods for predicting AQI in smart cities has demonstrate by comparing various machine learning algorithm such as XGBoost, decision tree regression, random forest regression and linear regression (35). The model was tested using the data of the cities Lucknow, Gurugram, Delhi, Ahmedabad and Mumbai. Decision tree regression method showed RMSE of [83.5523], MSE of [6980.99] and MAE of [37.1321]. At the same time, it was noticed that XGBoost regression method showed RMSE of [62.3231], MSE of [3884.17] and MAE of [28.6584].

The results of research which applied XGboost, neural network and random forest for prediction of AQI revealed that XGBoost outperformed other models rendering an RMSE value of [32.6], R^2 value of [0.942] and MAE value of [18.98] (22).

Yet another research was conducted in China in which two indices RMSE and R2 were used for examining the variance in predicting AQI in Beijing (36). The techniques adapted were random forest regression, support vector regression and obtained the values obtained were [RMSE = 83.6716, R2= 0.8401] and [RMSE = 94.4918 and R2=0.9760].

It can be identified that the proposed ensemble model is suitable for predicting AQI in Indian cities. Table 8 presents a comparison of existing approaches with the proposed approach in terms of its performance.

Table 8: Findings of Performance Evaluation

Findings	Accuracy rate	References
XGBoost has outperformed well	XGBoost achieved 94.2 percent accuracy	(22)
AQI prediction using XGBoost, AdaBoost and K-nearest neighbors are estimated using RMSE, MAPE and MAE metrics	XGBoost showed 60 percent accuracy	(34)
XGBoost regression method performed well	XGBoost regression achieved 87 percent accuracy	(35)
Random forest regression, support vector regression are adopted in their research	Random forest showed 84 percent accuracy	(36)
Gradient boosting and Random Forest model created in python exhibits promising accuracy level in evaluating AQI	Proposed model rendered 99.92 percent accuracy	Proposed model

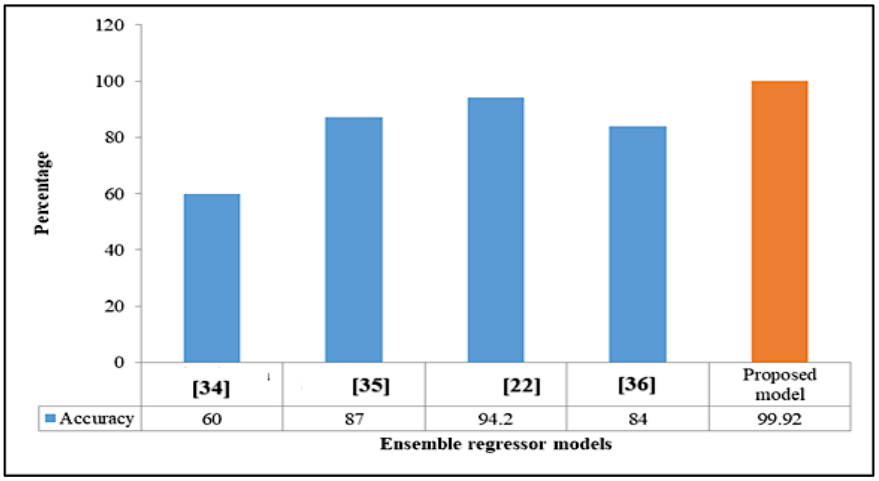

Figure 18: Comparative Analysis of Models Using Existing Approach

Figure 18 illustrates performance evaluation metrics. In the machine learning field, Random Forest and Gradient Boosting can be considered effective ensemble learning techniques, when it comes to classification and regression problems. However, both the methods take different

techniques to build the model. When Gradient Boosting builds trees one after the other, attempting to fix the errors of its predecessors, Random Forest builds a few independent decision trees simultaneously and bring their predictions altogether. Though they have specific things in common, both techniques differ largely in how they build and combine different decision trees. Gradient boosting trains trees in a sequential way and every new tree is measured by the algorithm by making use of the prior ensemble's residual errors. On the other hand, every tree is trained separately in Random Forests.

Conclusion

Gradient boosting method is a numerical optimization technique, which is essentially intended at acquiring an additive model that essentially reduces the loss function. In this context, gradient boosting repeatedly includes a new decision tree, which could lessen the loss function at each stage to the greatest extent possible. To be more precise, at every step, a new decision tree is added to the current and previous model to update the remaining trees. The gradient boosting algorithm, in general, builds each tree considering a subset of the training data and as well random features. Random forest makes use of a technique, i.e. bootstrap aggregating. In boosting techniques-based ensemble models, multiple decision trees are trained independently on different random subsets of data, while gradient boosting include new tree to each iteration, to reduce the residual errors. Advantages of random forest includes less proneness with regards to overfitting, handles missing data and above all easy to implement. Contrarily, the advantages of Gradient Boosting include flexibility, high predictive power and handling of imbalanced data. To conclude, the gradient boosting and Random Forest model created in python exhibits promising performance with approximately 85% (random forest) and 99% (gradient boosting) in evaluating air quality index (AQI). Similarly, the MSE scores of the gradient boosting model rapidly improved when compared to the non-improvised random forest model which shows that gradient boosting is a better fit than random forest model for AQI. Both these techniques hold enormous potential for improving air quality monitoring and playing an important role in addressing environmental complications.

Like every other research, the current research is also subject to several limitations. In the current research the training and testing is done as 80:20 split for the random forest approach. The training and testing of the random forest could be done using a time series-based split in the future to enhance the performance of the random forest approach. A detail elucidation of patterns and impact of each pollutant on each city could be studied in the future by employing advanced machine learning models. This research could be extended further by developing advanced models like transformer models and deep learning models and the same dataset could be tested against these models to enhance the efficiency of the prediction of air quality index.

Abbreviations

APSI: Air Pollution Standard Index, AQI: Air Quality Index, CPCB: Central Pollution Control, Board, CPU: Central Processing Unit, GB: Gradient Boost, MAE: Mean Absolute Error, MSE: Mean Squared Error, RAM: Random Access Memory, RF:Random Forest, RMSE: Rooted-mean square error.

Acknowledgement

None.

Author Contributions

R Vetri Selvi: conceptualization, data collection, analysis, implementation of the model, writing and manuscript preparation, R Satish Babu: Project Supervision.

Conflict of Interest

The authors declare no potential conflict of interest.

Data Availability

Secondary data has been used in this research and the data source has been mentioned in the manuscript in reference number (30).

Declaration of Artificial Intelligence (AI) Assistance

The authors declare that no generative AI was used in the writing process.

Ethics approval

The researchers obtained ethical approval from the University Ethics Committee.

Funding

None.

References

1. Bhuyan A, Bordoloi T, Debnath R, *et al*. Assessing AQI of air pollution crisis 2024 in Delhi: Its health risks and nationwide impact. Discover Atmosphere. 2025;3:13. doi: 10.1007/s44292-025-00041-x
2. Natarajan SK, Shanmurthy P, Arockiam D, *et al*. Optimized machine learning model for air quality index prediction in major cities in India. Sci Rep. 2024;14:6795. doi: 10.1038/s41598-024-54807-1
3. Hemamalini RR, Rajasekaran V, Balusamy S, *et al*. Air quality monitoring and forecasting using smart drones and recurrent neural network for sustainable development in Chennai city. Sustain Cities Soc. 2022;85:104077. doi: 10.1016/j.scs.2022.104077
4. Arulprakasajothi M, Chandrasekhar U, Yuvarajan D, *et al*. An analysis of the implications of air pollutants in Chennai. Int J Ambient Energy. 2020;41(2):209–13. doi: 10.1080/01430750.2018.1443504
5. Navasakthi S, Pandey A, Bhari JS, *et al*. Significant variation in air quality in South Indian cities during COVID-19 lockdown and unlock phases. Environ Monit Assess. 2023;195:772. doi: 10.1007/s10661-023-11375-7
6. Singh RP, Chauhan A. Impact of lockdown on air quality in India during COVID-19 pandemic. Air Qual Atmos Health. 2020;13:921–8. doi: 10.1007/s11869-020-00863-1
7. Gokul PR, Mathew A, Bhosale A, *et al*. Spatio-temporal air quality analysis and PM2.5 prediction over Hyderabad city, India using artificial intelligence techniques. Ecol Inform. 2023;76:102067. doi: 10.1016/j.ecoinf.2023.102067
8. Wang J, Ji L, Li X, *et al*. A hybrid air quality index prediction model based on CNN and attention gate unit. IEEE Access. 2022;10:113343–54. doi: 10.1109/ACCESS.2022.3217242
9. Zhan H, Zhu X, Hu J. A probabilistic forecasting approach for air quality spatio-temporal data based on kernel learning method. Appl Soft Comput. 2023;132:109858. doi: 10.1016/j.asoc.2022.109858
10. Chhikara P, Tekchandani R, Kumar N, *et al*. Federated learning and autonomous UAVs for hazardous zone detection and AQI prediction in IoT environment. IEEE Internet Things J. 2021;8(20):15456–67. doi: 10.1109/JIOT.2021.3074523
11. Samad A, Garuda S, Vogt U, *et al*. Air pollution prediction using machine learning techniques – an approach to replace existing monitoring stations with virtual monitoring stations. Atmos Environ. 2023;310:119987. doi: 10.1016/j.atmosenv.2023.119987
12. Yafouz A, Ahmed AN, Zaini N, *et al*. Ozone concentration forecasting based on artificial intelligence techniques: a systematic review. Water Air Soil Pollut. 2021;232(2):79. doi: 10.1007/s11270-021-04989-5
13. Zaini N, Ean LW, Ahmed AN, *et al*. A systematic literature review of deep learning neural network for time series air quality forecasting. Environ Sci Pollut Res. 2022;29(4):4958–90. doi: 10.1007/s11356-021-17442-1
14. Zhang D, Woo SS. Real-time localized air quality monitoring and prediction through mobile and fixed IoT sensing network. IEEE Access. 2020;8:89584–94. doi: 10.1109/ACCESS.2020.2993547
15. Rad AK, Nematollahi MJ, Pak A, *et al*. Predictive modeling of air quality in the Tehran megacity via deep learning techniques. Sci Rep. 2025;15:1367. doi: 10.1038/s41598-024-84550-6
16. Aggrawal S, Bhushan B. Machine learning for air pollution. Proc 2nd Int Conf Emerg Trend Communic Network Technol (ViTECoN); Vellore, India. 2023:1–6. doi: 10.1109/ViTECoN58111.2023.10157028
17. Hire M, Yadav P, Pandey S, *et al*. Indian air quality forecasting using ensemble methods for early warning systems. Int Res J Eng Technol. 2025;12(3): 760–4. e-ISSN: 2395-0056 https://www.irjet.net/archives/V12/i3/IRJET-V12I3117.pdf
18. Yang J, Tian Y, Wu CH. Air quality prediction and ranking assessment based on Bootstrap-XGBoost algorithm and ordinal classification models. Atmosphere. 2025;15(8):925. doi: 10.3390/atmos15080925
19. Patil A, Patil P. Air quality index prediction using machine learning. Int J Adv Comput Theory Eng. 2025;14(1):43–7. doi: 10.65521/ijacte.v14i1.211
20. Mishra D, Sahu A, Naman S. Implementation of boosting algorithms in predicting air quality index of South Indian cities. Eng Innov. 2025;14:99–115. doi: 10.4028/p-wcntd6
21. Diallo AA, Nderu L, Malenje BM, *et al*. Enhancing outlier detection in air quality index data using a stacked machine learning model. Eng Rep. 2024;6(11):e12936. doi: 10.1002/eng2.12936
22. Dao T, Nhat V, Trung H, *et al*. Analysis and prediction for air quality using various machine learning models. Proc Seventh Int Conf Research in Intell Comput Eng. 2022:89–94. doi: 10.15439/2022R03
23. Sapari A, Hadiana A, Umbara F. Air quality classification using extreme gradient boosting (XGBoost) algorithm. Innov Res Inform. 2023;5(2): 44–51. doi: 10.37058/innovatics.v5i2.8444
24. Oumoulylte M, Allaoui A, Farhaoui Y, *et al*. Efficient air quality prediction models based on supervised machine learning techniques. E3S Web Conf. 2025;632:02012. https://doi.org/10.1051/e3sconf/202563202012
25. Choudhary B, Pandey B. Predicting air quality index in Bhopal, India through machine learning approaches. Afr J Biomed Res. 2024;27(5S):294–304. https://doi.org/10.53555/AJBR.v27i5S.5391

26. Alzubi F, Al-Rawabdeh A, Almagbile A. Predicting air quality using random forest: A case study in Amman–Zarqa. Egypt J Remote Sens Space Sci. 2024;27(3): 604–13.
doi: 10.1016/j.ejrs.2024.07.004
27. Sathvika G, Poojitha P, Rakesh K, *et al*. Air quality prediction using random forest regression. J Emerg Technol Innov Res. 2024;11(6):7–84.
ISSN: 2349-5162
28. Setiawan A, Wibowo U, Mubarok A, *et al*. Random forest algorithm to measure the air pollution standard index. Knowl Eng Data Sci. 2024;7(1):86–100.
doi: 10.17977/um018v7i12024p86-100
29. Singh S, Yadav A, Kumar A. Prediction of air pollution using random forest. Ann Romanian Soc Cell Biol. 2021;25(4):19314–22.
http://annalsofrscb.ro/index.php/journal/article/view/8536
ISSN: 1583-6258
30. Panday A. Air quality data in India (2015–2024) [dataset]. Retrieved: May 5, 2025.
https://www.kaggle.com/datasets/ankushpanday1/air-quality-data-in-india-2015-2024
31. Singhal M, Verma M. Air pollution trends in metropolitan cities of India: Tales of the future generation. NMO J. 2023;17(1):21–5.
doi: 10.53772/NMO.2023.17106
32. Salman HA, Kalakech A, Steiti A. Random forest algorithm overview. Babylon J Machine Learn. 2024:69-79.
doi: 10.58496/BJML/2024/007
33. Natekin A, Knoll A. Gradient boosting machines, a tutorial. Front Neurorobot. 2013;7:21.
doi: 10.3389/fnbot.2013.00021
34. Kothandaraman D, Praveena N, Varadarajakumar K, *et al*. Intelligent forecasting of air quality and pollution prediction using machine learning. Absorpt Sci Technol. 2022;22:5086622.
doi: 10.1155/2022/5086622
35. Sharma G, Khurana S, Saina N, *et al*. Comparative analysis of machine learning techniques in Air Quality Index (AQI) prediction in smart cities. International Journal of System Assurance Engineering and Management. 2024;15:3060-3075.
doi: 10.1007/s13198-024-02315-w
36. Liu H, Li Q, Yu D, *et al*. Air quality index and air pollutant concentration prediction based on machine learning algorithms. Applied Sciences. 2019;9(19):4069.
doi: 10.3390/app919406

How to Cite: Selvi RV, Babu RS. Application of Mathematical Models for Prediction of Air Quality of Select Indian Cities Through Ensemble Learning Approach. Int Res J Multidiscip Scope. 2026; 7(2): 1310-1327.
DOI: 10.47857/irjms.2026.v07i02.08199