CSE445 Report

Air Quality Index Prediction

Mohammed Arif Mainuddin 2211578042

Fuwad Hasan 2211247042

Riazul Zannat 2211199042

Najifa Tabassum 2211578042

Faculty: Riasat Khan

Assistant Professor

ECE Department

Spring 2025

Date: 01/05/2025

Individual Contribution Table

Section	Contributing member name	
IEEE Word/LaTEX formatting	Fuwad	
Grammarly check	Najifa & Fuwad	94%
Abstract	Zannat	
Keyword	Zannat	
Introduction motivation	Zannat	
Paper review1	Najifa	[3]
Paper review2	Zannat	[10]
Paper review3	Arif	[11]
Paper review4	Fuwad	[12]
Paper review5	Najifa	[1]
Paper review6	Arif	[2]
Introduction last paragraph	Zannat	
Proposed system(Dataset & preprocessing)	Arif & Najifa	
Proposed system(Model description)	LR- Arif	
	ADABoost- Arif	
	KNN- Zannat	
	Bagging- Zannat	
	RF- Najifa	
	GradientBoosting- Najifa	
	DecisionTree- Fuwad	
	Stacking- Fuwad	
Results & Discussion	Fuwad & Arif	
Figures & Table title formatting	Arif & Najifa	
Conclusion	Arif	
Equation Formatting	Fuwad & Zannat	
References formatting in IEEE format	Fuwad & Najifa	
Future Work	Zannat	

Air Quality Index Prediction using Machine Learning models

Fuwad Hasan
Electrical and Computer Engineering
North South University)
Dhaka, Bangladesh
fuwad.hasan@northsouth.edu

Najifa Tabassum
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
najifa.tabassum@northsouth.edu

Mohammed Arif Mainuddin
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mohammed.mainuddin@northsouth.edu

Riazul Zannat
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
riazul.zannat@northsouth.edu

***Abstract*—Air pollution remains a major environmental and public health concern, especially in urban areas. The aim of this project is to predict the Air Quality Index (AQI) using supervised machine learning techniques on real-world data from the Australian Capital Territory (ACT) Government's open data portal. The dataset was collected from three monitoring stations: Monash, Florey, and Civic, which includes 22 air pollutant-related features and 343940 records. Several machine learning models were evaluated and used following data preprocessing steps, which include feature scaling, feature selection, and handling null values etc. In this project, each model was assessed by performance metrics suitable for continuous numerical values. Among all proposed models, the Bagging Regressor was the best-performing model, containing a 99.95% R2 score along with some enhanced scores such as MAE=1.5352, RMSE=3.012. Furthermore, we have used lime on our best-performing model, LIME (Local Interpretable Model-agnostic Explanations), providing valuable insights into feature contributions. The results demonstrate the importance of machine learning in AQI prediction and offer a data-driven foundation for environmental policy and health advisory systems.**

Keywords—air quality index, machine learning, environmental data, regression models, bagging regressor, LIME, air pollution prediction, model evaluation

I Introduction

Think of the Air Quality Index (AQI) as a single number that sums up how clean or polluted the air is, based on measurements of key contaminants like particulate matter and gases [1]. When AQI readings climb, we can see the real-world fallout , acid rain, choking smog, and all the havoc they wreak on our environment [2]. Breathing tainted air takes a heavy toll on our bodies, contributing to everything from lung infections and chronic heart conditions to strokes, cancer, and even early death [1] [3].

Here in Bangladesh, that hidden health crisis is hard to ignore: in 2019 alone, air pollution is estimated to have claimed around 123,000 lives [1] [4]. It doesn't stop at human suffering; Dhaka's skyline of rolling buses and idling factories translates into over $800 million lost each year in medical bills and lost productivity [5]. Raising public awareness and encouraging people to use renewable energy sources will help to improve air quality [6]. At the same time, spreading the word about cleaner energy options and harnessing machine learning and even time-tested tools like gravimetric filters or gas chromatography, gives us both awareness and the data we need to act [8] [7]. In short, cutting pollution through smarter technology and public engagement is our best bet for healthier communities and a steadier economy [2] [4] [6].

This paragraph provides in-depth reviews of the most notable models for estimating the Air Quality Index(AQI) and the concentration of various pollutants through machine learning algorithms such as regression techniques.

Al-Eidi et al. [3] predicted the "Air Quality Index" using various regression approaches based on machine learning algorithms. The dataset comprised 103,205 records, featuring data from monitoring stations across ten diverse sites in Pune City. The authors considered three types of regression models to predict air quality, and the decision tree regression approach was superior to all other models. The decision tree evaluation results were MAE = 1.97, RMSE = 9.94, and R2 score 98.82 during training. Additionally, authors handled outlier values by replacing them with lower and upper boundary values.

Gupta and colleagues [10] predicted the Air Quality Index (AQI) across four major Indian cities using supervised machine learning techniques. The authors used the dataset, which contains AQI's hourly and daily records across four major cities in India from 2015 to 2020. The dataset was imbalanced, so they applied the synthetic minority oversampling technique (SMOTE) to balance it. Evaluation metrics used here are R-SQUARE, MSE, RMSE, MAE, and accuracy. The authors applied different models. Among them, the gradient boosting technique achieved the highest accuracy of 93.6% and an R-SQUARE value of 0.94, making it the most efficient model.

Ram et al. [11] implemented an optimized parameter on the atmosphere to determine air quality by employing multiple Machine Learning models. There were 108,035 rows and 16 columns of samples in the category. A visual representation was used to show the strength and direction between variables. The XGBoost model attained the best accuracy of 96.50%.

Maltare and Vahora [12] present a comparative study of three forecasting approaches. Seasonal ARIMA, SVM with various kernels, and LSTM for predicting the Air Quality Index (AQI) in Ahmedabad based on hourly pollutant data from CPCB and SAFAR spanning January 2015 to January 2021. They preprocessed the data by removing outliers, imputing missing values, and selecting key pollutant indices for modeling. Their experiments were evaluated with R2 Score, MSE, and RMSE metrics. From their obtained results, SVM with an RBF kernel achieves the highest accuracy (R2 = 0.9989, RMSE = 4.94), outperforming both SARIMA and LSTM models. The authors conclude that RBF-SVM is the most effective method for AQI prediction in Ahmedabad, and recommend applying their preprocessing and modeling framework to other regions.

Natarajan et al. [1] predicted “Air Quality Index” using an optimized machine learning model which combines Grey Wolf Optimization(GWO) along with decision tree and some conventional machine learning models for accurate prediction of air quality in major cities of India, dataset collected from the Kaggle repository. The proposed model (GWO) exhibited enhanced performance compared to conventional machine learning models. GWO-DT’s highest accuracy was for Hyderabad city, which was 97.66

Gupta et al. [2] presented a comparative Air Quality Index prediction analysis using machine learning models. Three different models were applied before and after using SMOTE. SMOTE is used for balancing the data. The authors concluded that the dataset with the SMOTE algorithm produced higher prediction accuracy.

The literature review emphasizes the effectiveness and limitations of using different machine learning models to predict the Air Quality Index.

We need a better air quality predictor because of the harmful impact on health and the economy. This project focuses on predicting the Air Quality Index (AQI) using various machine learning models, including Linear Regression (LR), K-Nearest Neighbors (KNN), Gradient Boosting (GB), Decision Tree (DT), Random Forest (RF), AdaBoost, Stacking, and Bagging. The dataset underwent comprehensive preprocessing, including feature selection, feature scaling, and handling of missing values, to ensure high-quality input. Each model was evaluated using key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R-squared (R^2) score. Default and optimized hyperparameter settings were tested to find the most effective configurations. Among all the models, the Bagging Regressor demonstrated the highest prediction accuracy, and its outputs were further interpreted using LIME (Local Interpretable Model-agnostic Explanations) to enhance model transparency.

The rest of this report is structured as follows: Section II describes the proposed system, detailing dataset preprocessing, machine learning models, and evaluation methods with relevant tables, figures, and flowcharts. Section III showcases the results and discussion, comparing model performance using evaluation metrics and figures, tables, and lime ai on the best-performing model. Section IV summarizes the report by summarizing the findings and suggesting future work.

II Proposed System

The proposed system analyzes and visualizes the Air Quality Index based on Monash and Florey Australian city datasets. This system will help us identify the pollution pattern and support forecasting future air quality. We solely used Python as the programming language, along with various tools for visualization and multiple machine learning models.

A. Dataset Details

The dataset used in this project contains 343940 instances featuring data monitoring in three cities- Monash, Florey, and Civic. We took the dataset from The Government of Australia’s website [14], and last updated the dataset on February 16, 2025. Our project focused on 22 features related to the air quality index. Some of the features are independent, and some of them are dependent. Features are: Name, GPS, DateTime, NO2, O3_1hr, O3_4hr, O3_8hr, CO, PM10 1 hr, PM2.5 1 hr, AQI_CO, AQI_NO2, AQI_O3_1hr, AQI_O3_4hr, AQI_O3_8hr, AQI_PM10, AQI_PM2.5, AQI_Site, Date,Time.

The independent features (NO2, O3_1hr, O3_4hr, O3_8hr, CO, PM10 1 hr, PM2.5 1 hr) are the core pollutants. AQI_Site is the ground truth. This dataset emphasizes hourly air quality monitoring data from monitoring stations. The key observations from the dataset are negative min values, which represent the presence of outliers in features like PM10_1hr, PM2.5, and C0. Count refers to missing entries in features. AQI_O3_8hr has the highest missing entries. Moreover, the median indicates pollution levels in the dataset. NO2 has the lowest pollution level.

TABLE I
DATASET DESCRIPTION SHOWING MINIMUM, MAXIMUM, AND AVERAGE VALUES FOR SELECTIVE FEATURES

Features	Minimum	Maximum	Average
NO2	-0.0010	0.1700	0.0045
O3_1hr	-0.0010	1.9160	0.0169
O3_4hr	-0.0010	0.1180	0.0172
O3_8hr	0.0000	0.1070	0.0170
CO	-0.1400	22.0000	0.2247
PM10 1hr	-55.000	2714.00	12.3410
PM2.5 1hr	-38.000	2496.00	8.1818
PM10	-2.0000	1216.00	12.2966
PM2.5	-4.0000	1296.00	8.1735

B. Dataset Preprocessing

Scientists use dataset preprocessing to check for erroneous data that could give insight into producing fall prediction, and balance the dataset if the dataset is imbalanced via various methods.

Duplicate Values: Initially, there were 3,43,940 samples and 22 features, with 20 duplicate samples. Since the duplicate sample number is much less than the whole dataset, we dropped the duplicate values, leaving the dataset with 3,43,920 samples and 22 features.

Null values: There were enormous null values in the entire dataset. Total null values were 1039753, and the AQI_O3_8hr had the maximum null values, around 174538. However, the AQI_Site had only 120 null values. The dataset might create biases if we replaced all null values except the AQI_Site with various statistical values. The median of the AQI_Site replaced the null values of AQI_Site, and the mean value of the AQI_CO replaced the null values of the AQI_CO. Finally, we removed all the remaining null values to prevent bias.

Feature Scaling: Scientists use feature scaling if the values of one feature are much smaller than those of any other feature. In our dataset, the values of NO2, CO, O3_1hr, O3_4hr, and O3_8hr are near zero; however, the values of other features like PM10, PM2.5, etc. are much larger than 0. Also, our dataset did not produce a Gaussian shape curve to use Standardization for feature scaling, so we used the min-max scaler technique to scale our dataset.

$$x_j^{(\text{scaled})} = \frac{x_j - \min_i(x_j^{(i)})}{\max_i(x_j^{(i)}) - \min_i(x_j^{(i)})} \tag{1}$$

where

- $x_j^{(i)}$ is the value of the j-th feature for the i-th sample,
- $\min_i(x_j^{(i)})$ and $\max_i(x_j^{(i)})$ are the minimum and maximum values of feature j over all samples,
- $x_j^{(\text{scaled})}$ is the resulting value scaled to the $[0, 1]$ range.

Feature selection: Feature selection is a convenient way to handle imbalanced data in a large dataset. There are many ways to apply feature selection, but the Pearson-Correlation method has been used in this project, as this method works best for regression-type datasets. To apply feature selection, Split the data into train-test data using an 80:20 ratio. Here, X = all the features after feature scaling, y = AQI_Site. Corr = X_train function was used to demonstrate the correlation between features. corr_features = correlation(X_train, 0.98), in this project, the threshold value of highly correlated features was 0.98, which means that if two features are 98% correlated, they can be claimed as identical, so one of them can be removed to improve the model's performance.

Here is the heatmap of correlated features to visualize the correlation between features.

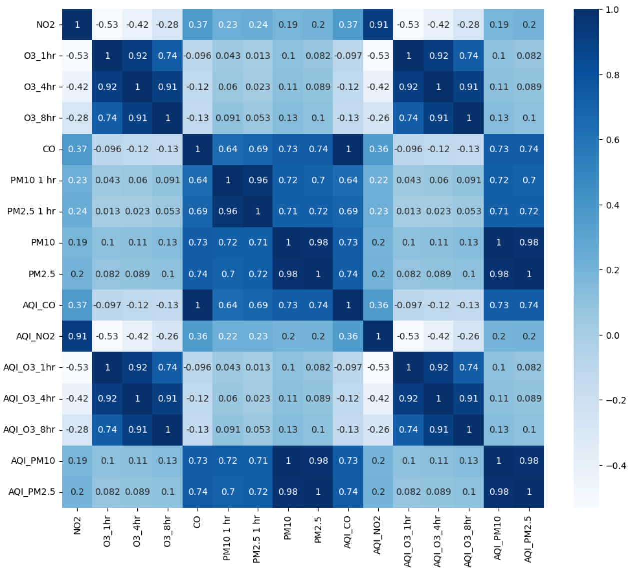

Fig. 1. Heatmap of feature correlation presenting correlation between features.

Handling Imbalanced Data: Since in this project the dataset contains regression-type data for all the features, and the target variable is continuous, the balancing methods such as SMOTE, random oversampling, or undersampling have not been applied. On the contrary, imbalanced data was handled by using robust regression models.

Exploratory Data Analysis: EDA was performed to analyze the data's underlying structure, identify outliers, and gain insights into relationships between all features vs AQI_Site. Various visualizations such as scatter plots, box plots, KDE plots, boxplots, etc, were used to explore feature density with the target variable.

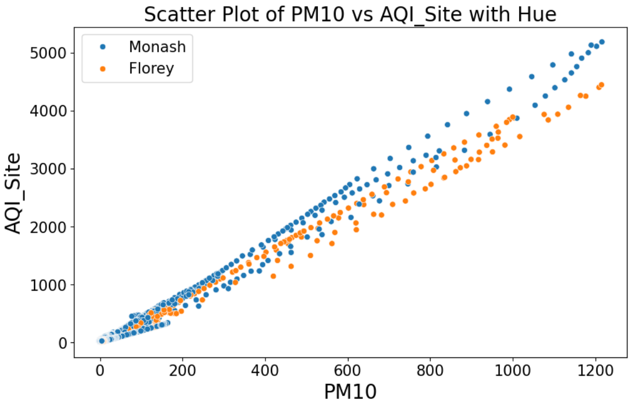

Fig. 2. A scatterplot containing PM10 vs. AQI_Site, where scatter plots define the spreadness or how much data points are scattered from each other. In the figure, Monash and Florey city data points are plotted, where data points are in a decent correlation with each other and are not highly scattered.

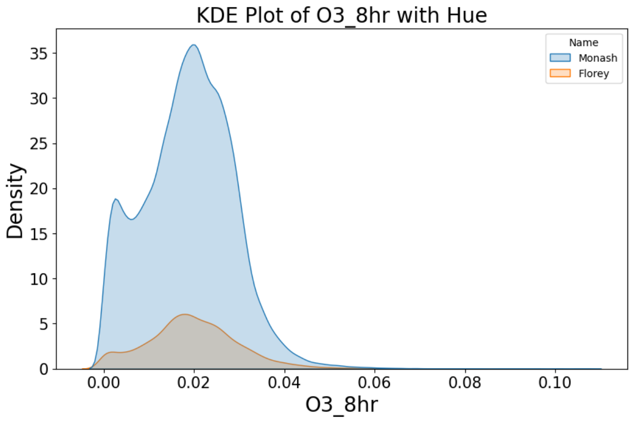

Fig. 3. KDE plot of O3_8hr(Ozone level over 8 hr) vs Density for two monitoring stations- Monash and Florey. Both distributions are right-skewed, where Monash seems to have a higher and sharper peak, and Florey has a flatter distribution.

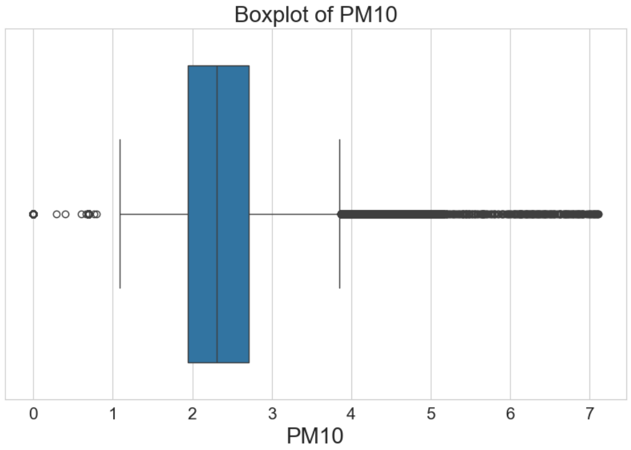

Fig. 4. explains more about outliers in the data, focusing on a specific feature like PM10. Median lies near the center of the box, but there are many outlier data points on the right side extending beyond 7.0, as well as a small number of outliers on the left side

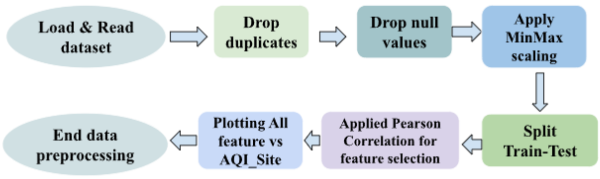

Fig. 5. visualizes the overall steps in this project to preprocess the dataset.

III Methodology

To predict the Air Quality Index, we evaluated eight different machine learning algorithms. We split the dataset into an 80/20 train–test partition using a fixed random state. Each model was first trained with its default hyperparameters, and then the hyperparameters were fine-tuned using GridSearchCV or RandomizedSearchCV. We evaluated all models using four metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). Their definitions are:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{2}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2, \tag{3}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}, \tag{5}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, and n is the number of observations.

A. Linear Regression

Linear regression analysis produces a best-fit line for univariate problems and a hyperplane for multivariate problems. It uses a cost function (such as Mean Squared Error) to optimize the hypothesis. This hypothesis, represented by the fitted line or hyperplane, is then used for prediction. One limitation of linear regression is that the model may suffer from overfitting with large and complex datasets, especially when the data has many features or noise.

Linear Regression Hypothesis:

$$h_\theta(x) \;=\; \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \;=\; \theta^\top x \tag{6}$$

Cost Function:

$$J(\theta) \;=\; \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \tag{7}$$

B. K-Nearest Neighbors (KNN) Regressor

K-Nearest Neighbors (KNN) regression is a non-parametric method that predicts the target value of a query point by averaging the outputs of its k closest training samples. In this project, the KNN model was first applied with default parameters and later optimized using GridSearchCV. The closeness between data points was measured using Euclidean distance, given by:

$$d\left(x, x^{(i)}\right) = \sqrt{\sum_{j=1}^{n}\left(x_j - x_j^{(i)}\right)^2} \tag{8}$$

where

- $d\left(x, x^{(i)}\right)$ is the Euclidean distance between the query point x and the i-th training example $x^{(i)}$.
- $x = (x_1, x_2, \ldots, x_n)$ is the n-dimensional feature vector of the query point.
- $x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right)$ is the n-dimensional feature vector of the i-th training example.
- n is the number of features (dimensions) in each vector.

C. Decision Tree Regressor

A decision tree is a recursive partitioning structure for decision support that employs a tree-like representation of decisions and their potential outcomes, such as utility, resource costs, and chance event outcomes. Decision Tree regression is a non-parametric, tree-based method that recursively partitions the feature space into disjoint regions.

D. RandomForest Regressor

A random forest is a meta-estimator that combines decision tree regressors on multiple sub-samples of the dataset. Random forest is a machine learning algorithm that helps to prevent overfitting issues, is less sensitive to noise/outliers, and gives an accurate prediction compared to other models. The model uses the best splitting strategy and controls sub-samples by defining the max_samples parameter. In this project, the model was used as a RandomForestRegressor from sklearn. It compares a before-and-after feature selection performance with default and optimized hyperparameters using RandomizedSearch CV.

$$\hat{y}(x) = \frac{1}{T}\sum_{t=1}^{T} h^{(t)}(x) \tag{9}$$

where

- $\hat{y}(x)$ is the ensemble prediction of the random forest for input feature vector x.
- T is the total number of trees in the forest.
- $h^{(t)}(x)$ prediction for the input x from t-th decision tree.

E. Gradient Boosting

Gradient boosting is an ensemble machine learning method that boosts weak learner models to strong learners. It performs sequential boosting with training models, where each new model tries to correct the error made by their predecessor. Boosting can improve accuracy by converting weak models into strong models. Also, gradient boosting is more robust as it updates the weights. It performed better in handling imbalanced data. In this project, the model was used as a GradientBoostingRegressor from sklearn. It compares a before-and-after feature selection performance with default and optimized hyperparameters using RandomizedSearch CV.

F. Adaboost

AdaBoost (Adaptive Boosting) is a machine learning process that uses ensemble techniques to train a model. It works by iteratively training weak learners, typically decision stumps, on the entire dataset while adjusting the weights of training samples. Initially, all samples are given equal weight by the model. After each iteration, the model increases the weights of misclassified samples so that the next learner can focus more on them. This process continues until the predefined number of learners is exhausted or it reaches the performance threshold. The key hyperparameters of AdaBoost include the number of estimators (learners) and the learning rate.

G. Stacking Regressor

Stacking regression is an ensemble technique that uses a meta-learner to integrate the predictions of several base learners. First, the training set is used to train M base regressors $\{h^{(1)}, \ldots, h^{(M)}\}$. The meta-learner, a second-level model g, uses their out-of-fold predictions on the training data (or held-out validation folds) as input features. At test time, the meta-learner generates the final output based on the predictions of the basis models.

$$\hat{y}(x) = g\bigl(h^{(1)}(x),\, h^{(2)}(x),\, \ldots,\, h^{(M)}(x)\bigr) \tag{10}$$

where

- $\hat{y}(x)$ is the final ensemble prediction for input feature vector x.
- $h^{(m)}(x)$ is the prediction of the m-th base regressor.
- M is the number of base regressors in the ensemble.
- $g(\cdot)$ is the meta-learner, typically a simple model (e.g., linear regression) trained on the base models' predictions.

H. Bagging Regressor

Bagging Regressor (Bootstrap Aggregating) is an ensemble learning technique that improves prediction accuracy by averaging outputs from multiple base regressors trained on randomly sampled subsets of the data. This approach reduces variance and helps prevent overfitting. In this project, the model was implemented using BaggingRegressor from sklearn. Ensemble, with performance evaluated before and after feature selection and optimized using RandomizedSearchCV. The final prediction $\hat{y}(x)$ for a new input x is calculated as:

$$\hat{y}(x) = \frac{1}{T}\sum_{t=1}^{T} h^{(t)}(x) \tag{11}$$

where

- $\hat{y}(x)$ is the aggregated prediction for feature vector x.
- T is the total number of base regressors in the ensemble.
- $h^{(t)}(x)$ is the prediction of the t-th base regressor trained on a bootstrap sample.

Here is the overall working sequence and flow chart for our proposed system and experiments:

- Loading the dataset
- Dataset Preprocessing
- Feature Scaling
- Feature Selection
- Train-Test Split
- Model Training
- Model Evaluation
- Model Comparison
- Select Best model for LIME XAI explanation

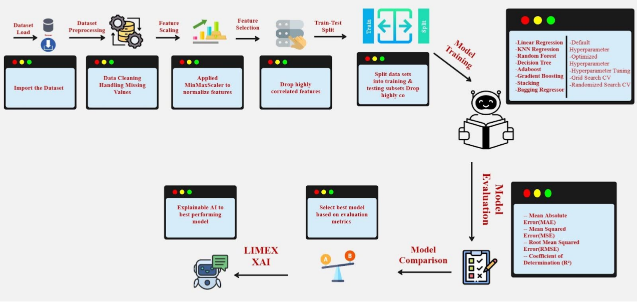
Dataset Load
Dataset Preprocessing
Feature Scaling
Feature Selection
Train-Test Split
Train
Split
Import the Dataset
Data Cleaning Handling Missing Values
Applied MinMaxScaler to normalize features
Drop highly correlated features
Split data sets into training & testing subsets Drop highly co
Model Training
-Linear Regression
-KNN Regression
-Random Forest
-Decision Tree
-Adaboost
-Gradient Boosting
-Stacking
-Bagging Regressor
-Default Hyperparameter
-Optimized Hyperparameter
-Hyperparameter Tuning
-Grid Search CV
-Randomized Search CV
Model Evaluation
-- Mean Absolute Error(MAE)
-- Mean Squared Error(MSE)
-- Root Mean Squared Error(RMSE)
-- Coefficient of Determination (R²)
Model Comparison
Select best model based on evaluation metrics
LIMEX XAI
Explainable AI to best performing model

Fig. 6. Working sequence of the proposed Air Quality Index Prediction system

IV Results and Discussion

In this section, we present the detailed results of our machine learning models for Air Quality Index Prediction. We have analyzed the performance of various models using different metrics and optimization techniques. The following tables illustrate the hyperparameter settings and performance results obtained throughout our experiments.

A. Hyperparameter Tuning

We conducted hyperparameter tuning using GridSearchCV and RandomizedSearchCV to optimize performance. Table II shows the range of hyperparameter values tested and the optimized values that yielded the best results.

TABLE II
HYPERPARAMETER VALUES RANGES FOR VARIOUS ML MODELS

Model	Hyperparameter Value Range	Optimized value
KNN	n_neighbors: [3, 5, 7, 9, 11], weights: [uniform, distance], algorithm: [auto, ball_tree, kd_tree, brute]	n_neighbors: 3, weights: distance, algorithm: ball_tree
Decision Tree	criterion: [squared_error, friedman_mse, absolute_error, poisson], max_depth: [None, 5, 10, 15, 20], min_samples_split: [2, 5, 10], min_samples_leaf: [1, 2, 4], max_features: [None, sqrt, log2]	criterion: squared_error, max_depth: 15, min_samples_split: 5, min_samples_leaf: 2, max_features: None
Random Forest	n_estimators: [50, 100, 200, 300], max_depth: [None, 5, 10, 15, 20], min_samples_split: [2, 5, 10], min_samples_leaf: [1, 2, 4], max_features: [None, sqrt, log2]	n_estimators: 50, max_depth: 10, min_samples_split: 5, min_samples_leaf: 4, max_features: None
Gradient Boosting	n_estimators: [50, 100, 200, 300], learning_rate: [0.01, 0.05, 0.1, 0.2], max_depth: [3, 5, 10, 15], min_samples_split: [2, 5, 10], min_samples_leaf: [1, 2, 4], subsample: [0.6, 0.8, 1.0], max_features: [None, sqrt, log2]	n_estimators: 200, learning_rate: 0.2, max_depth: 5, min_samples_split: 2, min_samples_leaf: 1, subsample: 0.8, max_features: None
AdaBoost	n_estimators: [50, 100, 200, 300], learning_rate: [0.01, 0.05, 0.1, 0.2, 0.5, 1.0], loss: [linear, square, exponential]	n_estimators: 100, learning_rate: 0.05, loss: exponential
Bagging	n_estimators: [10, 50], max_samples: [0.7, 1.0], max_features: [0.7, 1.0], bootstrap: [True, False]	n_estimators: 50, max_samples: 0.7, max_features: 1.0, bootstrap: False

B. Performance Analysis

We evaluated each model using multiple performance metrics, as discussed in the methodology section. Table III shows the performance metrics of various models with default hyperparameters.

After hyperparameter optimization, we observed significant improvements in model performance, as shown in Table IV.

TABLE III
PERFORMANCE METRICS OF VARIOUS ML MODELS WITH DEFAULT HYPERPARAMETERS

Model	MAE	MSE	RMSE	R^2 Coefficient
Linear Regression	6.8273	90.6845	9.5228	0.9953
KNN	6.2241	236.8677	15.3905	0.9878
Decision Tree	1.6433	16.7814	4.0965	0.9991
Random Forest	1.5727	**9.4914**	**3.0808**	**0.9995**
Gradient Boosting	3.0513	18.3982	4.2893	0.9991
Adaboost	15.7799	472.0011	21.7256	0.9757
Stacking	2.1676	13.5774	3.6848	0.9993
Bagging	**1.6196**	11.9155	3.4519	0.9994

TABLE IV
PERFORMANCE METRICS OF VARIOUS ML MODELS WITH OPTIMIZED HYPERPARAMETERS

Model	MAE	MSE	RMSE	R^2 Coefficient
Linear Regression	6.8273	90.6845	9.5228	0.9953
KNN	5.9972	190.1527	13.7896	0.9902
Decision Tree	1.8538	23.3133	4.8284	0.9988
Random Forest	2.6358	19.3719	4.4013	0.9990
Gradient Boosting	2.0008	8.7428	2.9568	**0.9995**
Adaboost	14.7044	448.8713	21.1866	0.9769
Stacking	0.325	**0.209**	**0.457**	0.813
Bagging	**1.5352**	9.0724	3.0120	**0.9995**

Finally, we performed feature selection to further enhance model performance. Table V presents the performance metrics after both hyperparameter optimization and feature selection.

TABLE V
PERFORMANCE METRICS OF VARIOUS ML MODELS WITH OPTIMIZED HYPERPARAMETERS AND FEATURE SELECTION

Model	MAE	MSE	RMSE	R^2 Coefficient
Linear Regression	6.8900	91.7683	9.5796	0.9953
KNN	6.0098	183.0589	13.5299	0.99056
Decision Tree	2.4200	17.8558	4.2256	0.9991
Random Forest	2.8006	19.4293	4.4079	0.9989
Gradient Boosting	2.3212	**10.5421**	**3.2469**	0.9994
Adaboost	14.7643	447.2555	21.1484	0.9769
Stacking	2.4633	15.8592	3.98237	0.9992
Bagging	**2.1049**	11.9149	3.4518	**0.9995**

TABLE VI
OVERALL PERFORMANCE COMPARISON OF VARIOUS ML MODELS (TAKING THE BEST MODELS)

Model	R^2 Score	MAE	MSE	RMSE
Bagging	**0.999532**	**1.535251**	**9.072430**	**3.012047**
Gradient Boosting	0.999457	2.321222	10.542126	3.246864
Stacking	0.999300	2.167566	13.577444	3.684758
Decision Tree	0.999080	2.420000	17.855823	4.225615
Random Forest	0.998998	2.800551	19.429292	4.407867
Linear Regression	0.995269	6.890008	91.768271	9.579576
KNN	0.990563	6.009776	183.058964	13.529928
AdaBoost	0.976944	14.764341	447.255506	21.148416

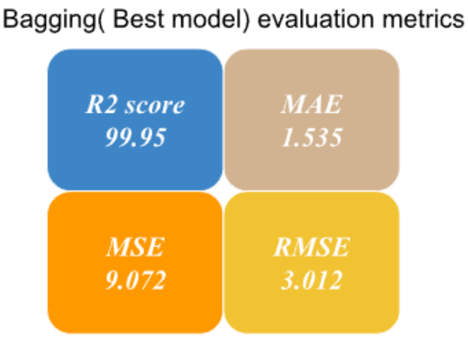

Fig. 7. Bagging Regressor model performance metrics after hyperparameter tuning

C. Discussion

From our experiments with these Machine Learning models and the results obtained, we can conclude that Bagging Regressor performs best for this task as it clearly has the best R^2 score and the lowest MAE, MSE and RMSE scores. The other models also performs good but AdaBoost has the lowest performance among all.

D. LIME Explainable AI

To understand how our Air Quality Index Prediction model makes a specific prediction, we used LIME (Local Interpretable Model-agnostic Explanations) [13]. LIME helps us to explain how the model makes predictions by highlighting the most important features that influence a specific prediction.LIME is applied to a test data instance from our dataset. We can visually see in Fig 8 the contribution of each feature into the final prediction as shown in the following figure:

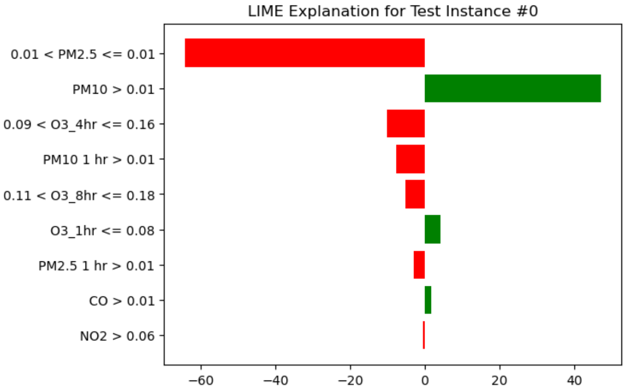

Fig. 8. LIME Explanation for an instance from our dataset shows individual contribution of each feature for the final predicted value.

TABLE VII
COMPARISON OF THE PROPOSED SYSTEM WITH EXISTING WORKS

Ref	Model	R^2 Score	Other Metrics
[3]	Decision Tree	98.82	MAE= 1.97, RMSE = 9.94
[10]	Gradient Boosting	93.6	Not mentioned in the paper
[10]	SVM	99.89	RMSE= 4.94
[11]	XGBoost	96.50	Not mentioned in the paper
This work	**Bagging**	**99.95**	MAE= 1.53,RMSE= 3.015

V Conclusion and Future Works

A. Conclusion

In conclusion, this project successfully implemented various machine-learning techniques for the air quality index based on Monash and Florey. Through thorough preprocessing, null value handling, feature scaling, and dropping highly correlated features, we applied various machine learning models, from which the best model was the Bagging Regressor, with the highest R2 score. This project strengthens our understanding of environmental data analysis and depicts how we can apply machine learning for the welfare of our nature and environment.

B. Future Work

Our proposed system can be improved by introducing deep learning models, which can be used for non-tabular data. Besides this, our current best model can be used to develop a user-friendly web or mobile application that would make it simple for regular people and local government agencies to monitor the air quality in real-time.

References

[1] S. K. Natarajan, P. Shanmurthy, D. Arockiam, B. Balusamy, and S. Selvarajan, "Optimized machine learning model for air quality index prediction in major cities in India," *Scientific Reports*, vol. 14, 2024. Retrieved from https://www.nature.com/articles/s41598-024-54807-1.

[2] N. S. Gupta, Y. Mohta, K. Heda, R. Armaan, B. Valarmathi, and G. Arulkumaran, "Prediction of Air Quality Index Using Machine Learning Techniques: A Comparative Analysis," *Journal of Environmental and Public Health*, Jan. 2023. Retrieved from https://onlinelibrary.wiley.com/doi/10.1155/2023/4916267.

[3] S. Al-Eidi, F. Amsaad, O. Darwish, Y. Tashtoush, A. Alqahtani, and N. Niveshitha, "Comparative analysis study for air quality prediction in smart cities using regression techniques," *IEEE Access*, vol. 11, pp. 115140–115149, 2023. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10274948.

[4] C. Liu, G. Pan, D. Song, and H. Wei, "Air Quality Index Prediction Using Machine Learning: A Deep Learning Perspective," *IEEE Access*, vol. 11, pp. 67086–67097, 2023.Retrieved from https://ieeexplore.ieee.org/document/10168889.

[5] B. Zhang, M. Duan, Y. Sun, Y. Lyu, Y. Hou, and T. Tan, "Air Quality Index Prediction in Six Major Chinese Urban Agglomerations: A Comparative Study of Single Machine Learning Model, Ensemble Model, and Hybrid Model," *Atmosphere*, vol. 14, 2023.Retrieved from https://www.mdpi.com/2073-4433/14/10/1478.

[6] A. Mishra and Y. Gupta, "Comparative analysis of Air Quality Index prediction using deep learning algorithms," *Spatial Information Research*, vol. 32, pp. 63–72, 2024.Retrieved from https://link.springer.com/article/10.1007/s41324-023-00541-1.

[7] M. Emeç and M. Yurtsever, "A novel ensemble machine learning method for accurate air quality prediction," *International Journal of Environmental Science and Technology*, vol. 22, pp. 459–476, 2025. Retrieved from https://link.springer.com/article/10.1007/s13762-024-05671-z.

[8] S. A. Aram, E. A. Nketiah, B. M. Saalidong, H. Wang, A.-R. Afitiri, A. B. Akoto, and P. O. Lartey, "Machine learning-based prediction of air quality index and air quality grade: a comparative analysis," *International Journal of Environmental Science and Technology*, vol. 21, pp. 1345–1360, Jun. 2023.Retrieved from https://link.springer.com/article/10.1007/s13762-023-05016-2.

[9] I. Essamlali, H. Nhaila, and M. El Khaili, "Supervised machine learning approaches for predicting key pollutants and for the sustainable enhancement of urban air quality: A systematic review," *Sustainability*, vol. 16, p. 976, 2024. Retrieved from https://www.mdpi.com/2071-1050/16/3/976.

[10] N. S. Gupta, Y. Mohta, K. Heda, R. Armaan, B. Valarmathi, and G. Arulkumaran, "Prediction of Air Quality Index Using Machine Learning Techniques: A Comparative Analysis," *Journal of Environmental and Public Health*, vol. 2023, pp. 1–26, Jan. 2023.Retrieved from https://onlinelibrary.wiley.com/doi/10.1155/2023/4916267.

[11] M. S. Ram, C. Reshmasri, S. Shahila, and J. V. P. Saketh, "Air Quality Prediction using Machine Learning Algorithm," in *2023 International Conference on Sustainable Computing and Data Communication Systems*, pp. 316–321, 2023.Retrieved from https://ieeexplore.ieee.org/document/10105063.

[12] N. N. Maltare and S. Vahora, "Air Quality Index prediction using machine learning for Ahmedabad city," *Digital Chemical Engineering*, vol. 7, p. 100093, 2023.Retrieved from https://www.sciencedirect.com/science/article/pii/S277250812300011X?via%3Dihub.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, 2016. Retrieved from https://arxiv.org/abs/1602.04938.

[14] Government of the Australian Capital Territory, "Air Quality Monitoring Data," *dataACT*. [Online]. Available: https://www.data.act.gov.au/Environment/Air-Quality-Monitoring-Data/94a5-zqnn/about_data. [Accessed: 16-Feb-2025].