2025 International Conference on Computing Technologies & Data Communication (ICCTDC)
Hassan, India. July 04 - 05, 2025

Analyzing Hybrid Machine Learning Performance Through Air Quality Prediction

1st Abhijeet Rajhans
Computer Science and Engineering
Manipal University Jaipur
Jaipur, India
abhijeetrajhans.ar@gmail.com

2nd Ayush Dubey
Computer Science and Engineering
Manipal University Jaipur
Jaipur, India
ayushdubey21jan@gmail.com

3rd Akshay Jadhav
Computer Science and Engineering
Manipal University Jaipur
Jaipur, India
akshay.jadhav@jaipur.manipal.edu

2025 International Conference on Computing Technologies & Data Communication (ICCTDC) | 979-8-3315-2798-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCTDC64446.2025.11158796

***Abstract*—This paper focuses on machine learning approaches to analyze performance metrics based on air quality index prediction. We know that air pollution is one of the most serious problems faced by mankind to date and has severe consequences. The need to comprehend pollution patterns and accurately predict pollution levels will help communities take informed measures to mitigate its impact. This study begins with traditional machine learning approaches as the preliminary analysis and progresses toward more advanced hybrid models. This entire methodology aims to reduce or minimize the biases and overfitting of just one model by utilizing the properties of several models, thus improving AQI prediction accuracy. We compare the models on MSE, MAE and R^2 metrics and propose two models as they display the best overall performance. Our proposed hybrid models Model1 and Model2 attain one of the best R^2 value of 0.851680 and 0.848379, and MSE values of 2200.807 and 2249.7899 respectively. Model1 is defined as RandomForestRegressor + ExtraTreesRegressor and Model2 as RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Method of Stacking + Averaging). We do need to further evaluate these models with hyperparameter tuning to see how they change the results, however, we believe that these two hybrids, especially the latter, has a greater ability to generalize and obtain a near accurate prediction.**

***Index Terms*—Air Quality Index (AQI), Air Pollution, Machine Learning, Prediction, Overfitting-Bias Reduction, Model1, Model2**

I. Introduction

Air pollution is one of the most serious problems in the world which results in environmental degradation, health damage and it also disturbs the economic stability of the world.

The cause of air pollution can be both human-made and natural. Human-made factors include industrial activities that release harmful gases such as sulphur dioxide, nitrogen oxides, and other volatile organic compounds into the air, urbanization and construction processes lead to the release of fine particulate matter ($PM_{2.5}$ and PM_{10}), fossil fuel-powered plants also release pollutants that affect nature in negative way, and waste management can also lead to the release of harmful chemicals in nature which affect nature in negative way. Natural factors include Wildfires, dust storms and volcanic eruptions which lead to release of large amount of harmful chemicals in air. Air Pollution also leads to multiple problems which includes Health Impacts like respiratory diseases, Environmental Effects like acid rain which damages the environmental cycles, leads to Global warming and also has multiple economic effects.

Air Quality Index (AQI) is a measuring and monitoring tool that enables us to understand air quality levels in proper way and take useful actions to protect the environment from further degradation. It is a method that maps air pollutants concentration to numerical value. The standard formula to calculate the AQI score is given as :

$$AQI = \frac{A_{\text{high}} - A_{\text{low}}}{B_{\text{high}} - B_{\text{low}}} (C - B_{\text{low}}) + A_{\text{low}} \tag{1}$$

Where:

- C: Concentration of the pollutant.
- B_{low}: Lower concentration breakpoint for pollutant.
- B_{high}: Upper concentration breakpoint for pollutant.
- A_{low}: AQI value corresponding to B_{low}.
- A_{high}: AQI value corresponding to B_{high}.

This formula is adapted from the guidelines provided by the Central Pollution Control Board (CPCB) [2].

According to the Ministry of Environment, Forest and Climate Change [3], AQI levels can be divided into multiple ranges. AQI level from 0 – 50 is considered good, 50 -100 is considered as satisfactory, 101 – 200 is considered moderately polluted, 201 – 300 is considered poor, 301 – 400 is considered very poor and 401+ is considered as severe.

II. Literature Survey

N. Srinivas Gupta et al. [4] Used regression algorithms SVR, RFR, and CatBoost along with SMOTE to handle dataset imbalance. The highest accuracy with 85.08% in New Delhi was exhibited by Catboost, and the RFR with the maximum accuracy at 93.74% for Hyderabad and 97.61% for Kolkata had lowest RMSE (0.0628 for Hyderabad, 0.0988 for Kolkata). Suresh Kumar Natarajan et al. [5] used a combination of Decision Tree (DT) and Grey Wolf Optimization (GWO). R^2 , RMSE and accuracy were the evaluation metrics used for this paper. The model GWO-DT performed better than SVR (90.34%), k-NN (90.51%), and Random Forest (92.75%) by achieving 97.66% accuracy in Hyderabad, 97.68% in Visakhapatnam, and an overall average of 94.25% in all cities.

S. A. Aram et al. [6] used classical statistical methods like multiple linear regression and autoregressive integrated

moving average (ARIMA) which were supported by machine learning models such as support vector regression (SVR), decision trees, and random forests to exhibit enhanced predictive accuracy. These include Random Forest (RF) with R^2 = 0.991, RMSE = 6.590, and MAE = 1.048. An ensemble stacking model that had multiple base models had a higher R^2 value of 0.981, an RMSE value of 6.253, and an MAE value of 3.040. Random Forest provided the best classification with ACC = 0.999, MCC = 0.998, and F1-score = 0.999 on the training dataset. Tanisha Madan et al. [7] compared several algorithms for predicting Air Quality Index (AQI). A Deep Belief Network, which predicted the levels of $PM_{2.5}$ in China, achieved an error rate of 1.6%. A neural network predicting air temperature and relative humidity achieved an accuracy of 99.56%. For $PM_{2.5}$, hybrid models, tree-based algorithms, and light gradient boosting achieved an accuracy greater than 99%. Logistic regression and autoregression models achieved a mean accuracy of 99.859% with a standard deviation of 0.000612. Sophisticated models like MLP and XGBoost showed high R-squared values for ozone (98.65%) and general pollutants (99.51%).

Donghyun Kim et al. [8] checked the validity of LSTM and DNN models to predict the air pollutants and the CAI over South Korea for 2015–2020. This model exhibited good behavior in terms of capturing both the peak values and the temporal variability, with low NRMSE (0.10–0.19), excellent correlation (CC: 0.85–0.91), and NSE (0.91–0.96). DNN had performed quite well in the CAI prediction with an NSE of 0.96, an NRMSE of 0.10, and a CC above 0.91. Abdulrazak H. Almaliki et al. [9] applied models such as fine decision tree (FDT), ensemble boosted tree (EBOT), and ensemble bagged tree (EBAT) to air quality data from 2016 to 2018. EBOT reached peak accuracy at 97.4% by successfully predicting 75 of 77 samples, while FDT and EBAT managed to predict 96.1% and 94.8%, respectively.

Saba Ameer et al. [10] employed regression methods to analyze the connection between PM2.5 levels. Random Forest regression surpassed the other methods by demonstrating exceptional efficiency and superior peak identification accuracy, achieving the lowest mean absolute error (6%–18%) and root mean square error (0.05–0.18). Random Forest demonstrated the highest accuracy and data handling capabilities in larger datasets. Elia Georgiana Dragomir et al. [11] employed k-nearest neighbour (k-NN) algorithms to categorize pollution levels and forecast the air quality index (AQI). Using the Weka tool and applying 10-fold cross-validation, the model attained an accuracy of 65.51% with a 0% error rate for 19 out of 29 instances. The mean absolute error was 0.3793 and the root mean squared error was 0.6695, indicating a correlation coefficient of 0.5614 that demonstrated a moderate level of predictive capability.

III. Methodology

A. Preliminary Analysis

We first begin the analysis of the raw dataset [1] and apply different machine learning models on it to get a fundamental picture of the model that can be best suited to represent the variations in the data.

TABLE I: MSE, MAE, and R^2 Values for Preliminary Analysis of ML Models

Model	MSE	MAE	R^2
Random Forest Regression	2240.708348	33.83487095	0.847297376
Decision Tree Regression	4828.530579	46.50561974	0.670939196
Extra Trees Regression	2155.036338	33.13777454	0.85313586
Gradient Boosting Regression	3230.897449	42.78215239	0.779816718
XGBoost Regression	2474.6667	36.64448793	0.831353286

From table I, we can see that Random Forest Regressor and Extra Trees Regressor have the best results as compared to other models. However, we need to perform further analysis on the dataset to justify the requirement of the base estimator.

We begin the process by imputing the missing values in the dataset. For this, we consider three methods.

1) K-Nearest Neighbors Imputation
2) Spline Interpolation
3) Exponential Moving Average Interpolation.

1) K-Nearest Neighbors Imputation: KNN imputation deals with missing data by finding the nearest neighbors based on metrics such as Manhattan or Euclidean distance. Statistical metrics (mean, median, weighted average) are used to compute these neighbors' values in an aggregated format.

2) Spline Interpolation: Spline interpolation estimates missing data points by fitting piecewise polynomial curves to data, producing smooth joins of known data points.

3) Exponential Moving Average: The Exponential Moving Average (EMA) approximates missing observations in time series by calculating exponentially weighted averages of historical observations, with weights exponentially declining for previous data points.

On applying the three imputation techniques, we notice that Spline interpolation, EMA, and KNN imputation, paired with the Extra Trees Regressor, provide outstanding performance. Spline interpolation perennially returns near-perfect results because of its capacity to identify sophisticated, non-linear patterns. EMA excels at time-series data, taking advantage of recent trends in order to precisely impute. KNN imputation, though more inconsistent, is effective for tree-based models, especially Extra Trees. Generally, Extra Trees Regressor is the most reliable option for all imputation techniques, guaranteeing high-quality data preparation and analysis.

Our work applies KNN imputation, spline interpolation, and EMA, which handle missing data in unique ways—KNN identifies local dependencies, spline models smooth non-linear trends, and EMA follows time-based patterns. We mitigated each's limitation by averaging their imputed values, which eliminated biases, noise, and outlier sensitivities. Ensemble stabilizes outlier values, improves dataset reliability, and reduces estimation errors. By averaging different approaches,

TABLE II: MSE, MAE, and R^2 Values Without ARIMA and PCA

Model	MSE	MAE	R^2
Random Forest Regressor	2289.040816	33.993810	0.845734
Extra Trees Regressor	2162.766831	33.113874	0.854244
Decision Tree Regressor	5041.232260	47.026810	0.660255
Gradient Boosting Regressor	3382.765168	43.693405	0.772024
Elastic Net	4710.990901	54.395034	0.682511
XGB Regressor	3403.896894	43.814823	0.770600
AdaBoost Regressor	4730.364681	56.092892	0.681205
HistGradient Boosting Regressor	2818.297212	39.523759	0.810066
LGBM Regressor	2791.925643	39.371543	0.811843

TABLE III: MSE, MAE, and R^2 Values With PCA but Without ARIMA

Model	MSE	MAE	R^2
Random Forest Regressor	2651.009729	37.434085	0.821339
Extra Trees Regressor	2500.128207	36.159287	0.831508
Decision Tree Regressor	5332.899129	49.856815	0.640598
Gradient Boosting Regressor	3719.268553	46.295754	0.749346
ElasticNet	5345.282812	59.615595	0.639763
XGB Regressor	3715.709547	46.275696	0.749586
AdaBoost Regressor	5527.899757	60.575652	0.627456
HistGradient Boosting Regressor	3133.664377	41.869041	0.788812
LGBM Regressor	3125.315571	41.714364	0.789374

TABLE IV: MSE, MAE, and R^2 Values With ARIMA but Without PCA

Model	MSE	MAE	R^2
Random Forest Regressor	2319.520367	34.196830	0.843679
Extra Trees Regressor	2192.128591	33.325424	0.852265
Decision Tree Regressor	5097.894186	47.001761	0.656436
Gradient Boosting Regressor	3405.135790	43.704796	0.770516
ElasticNet	4732.445569	54.374528	0.681065
XGB Regressor	3430.933300	43.922010	0.768778
AdaBoost Regressor	4725.141544	55.978841	0.681557
HistGradient Boosting Regressor	2832.923729	39.536245	0.809080
LGBM Regressor	2813.196495	39.446511	0.810409

TABLE V: MSE, MAE, and R^2 Values With Both PCA and ARIMA

Model	MSE	MAE	R^2
Random Forest Regressor	2665.965623	37.452165	0.820331
Extra Trees Regressor	2519.848722	36.250975	0.830179
Decision Tree Regressor	5580.141149	51.191490	0.623936
Gradient Boosting Regressor	3717.899831	46.173920	0.749438
ElasticNet	5353.394378	59.579584	0.639217
XGB Regressor	3723.343660	46.239222	0.749071
AdaBoost Regressor	5531.131879	60.480375	0.627238
HistGradient Boosting Regressor	3150.714465	41.744568	0.787663
LGBM Regressor	3129.670829	41.776195	0.789081

the imputed dataset is more generalizable to machine learning models, enhancing predictive accuracy. Therefore, averaging these techniques provides a solid, accurate, and balanced dataset for analysis.

B. Model Selection for Base Estimator

Under this section, we shall use four methodologies for the analysis

1) Without Implementation of Auto-Regressive Integrated Moving Average and Principal Component Analysis
2) Without implementation of Auto-Regressive Integrated Moving Average but with implementation of Principal
3) Without implementation of Principal Component Analysis but with implementation of Auto-Regressive Integrated
4) With implementation of both Principal Component Analysis and Auto-Regressive Integrated Moving Average

From tables II, III, IV, and V, it is clearly evident that both Random Forest Regressor and Extra Trees Regressor lead the charts with the best performances, with Extra Trees Regressor consistently making a lead. Due to this, we declare Extra Trees as the L1 (base) Estimator and ARIMA + Extra Trees Regressor as the L1H (Hybrid) Estimator. It is to note that the L1H estimator, altogether, shall be treated as the base estimator hereafter. It is also to note that the p, d, q values in the ARIMA function were taken as 1, 1, 1 by default.

C. Development of Hybrid Machine Learning Models (Base ARIMA)

Under this section, we shall use only two methodologies for the analysis:

1) With implementation of both Principal Component Analysis and Base Auto-Regressive Integrated Moving Average (Table VI)
2) Without implementation of Principal Component Analysis but with implementation of Base Auto-Regressive Integrated Moving Average (Table VII)

D. Development of Hybrid Machine Learning Models (Adjusted ARIMA)

Under this section, we shall again use only two methodolies for the analysis:

1) With implementation of both Principal Component Analysis and **Adjusted** Auto-Regressive Integrated Moving Average (Table VIII)
2) Without implementation of Principal Component Analysis but with implementation of **Adjusted** Auto-Regressive Integrated Moving Average (Table IX)

From Table VI, we have defined the RandomForestRegressor + ExtraTreesRegressor model as the **Model1 Hybrid Estimator (Model1)**. Similarly, we have named the RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Method of Stacking + Averaging) model as the **Model2 Hybrid Estimator (Model2)**. In addition, the

TABLE VI: With implementation of both Principal Component Analysis and **Base** Auto-Regressive Integrated Moving Average

Models	MSE	MAE	R^2
RandomForestRegressor + ExtraTreesRegressor	2549.306199	36.578816	0.828239
RandomForestRegressor + XGBRegressor + ExtraTreesRegressor	2639.897147	37.578221	0.822089
XGBRegressor + ExtraTreesRegressor	2684.847335	38.026955	0.819059
XGBRegressor + ElasticNet + ExtraTreesRegressor	3026.043515	41.911815	0.796065
ElasticNet + ExtraTreesRegressor	3189.930823	43.678179	0.785020
LGBMRegressor + ExtraTreesRegressor	2721.859520	38.466719	0.816565
XGBRegressor + LGBMRegressor + ExtraTreesRegressor	2727.996772	38.475579	0.816151
XGBRegressor + Ridge Regression + ExtraTreesRegressor	2938.695058	41.072952	0.801952
RandomForestRegressor + ElasticNet + ExtraTreesRegressor	2974.811080	42.085028	0.799518
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor	3180.544025	44.090279	0.785653
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor (Stacking)	2767.881327	37.796816	0.813463
AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor	3315.988025	46.407616	0.776525
RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Stacking + Averaging)	2688.450505	37.777324	0.818817
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Averaging)	3285.031192	46.250383	0.778611
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Weighted Average)	2988.266568	43.194557	0.798611
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Stacking)	2888.609015	38.833785	0.805327

TABLE VII: Without implementation of Principal Component Analysis but with implementation of Auto-Regressive Integrated Moving Average

Models	MSE	MAE	R^2
RandomForestRegressor + ExtraTreesRegressor	2215.234010	33.494347	0.850708
RandomForestRegressor + XGBRegressor + ExtraTreesRegressor	2318.042727	34.792586	0.843780
XGBRegressor + ExtraTreesRegressor	2371.592363	35.454507	0.840171
XGBRegressor + ElasticNet + ExtraTreesRegressor	2777.856356	39.948153	0.812791
ElasticNet + ExtraTreesRegressor	2992.109684	42.018957	0.798352
LGBMRegressor + ExtraTreesRegressor	2405.994961	35.853801	0.837852
XGBRegressor + LGBMRegressor + ExtraTreesRegressor	2381.484952	35.665504	0.839504
XGBRegressor + Ridge Regression + ExtraTreesRegressor	2655.406374	38.970245	0.821044
RandomForestRegressor + ElasticNet + ExtraTreesRegressor	2639.953829	38.669667	0.822085
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor	2833.028876	41.295299	0.809073
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor (Stacking)	2348.481509	34.161170	0.841728
AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor	2960.242277	43.845573	0.800500
RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Stacking + Averaging)	2278.468564	34.343329	0.846447
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Averaging)	2951.927579	43.687419	0.801060
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Weighted Average)	2663.191932	40.559908	0.820519
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Stacking)	2449.963620	34.984250	0.834889

TABLE VIII: With implementation of both Principal Component Analysis and **Adjusted** Auto-Regressive Integrated Moving Average

Models	**MSE**	**MAE**	**R^2**
RandomForestRegressor + ExtraTreesRegressor	2545.607968	36.572312	0.828443
RandomForestRegressor + XGBRegressor + ExtraTreesRegressor	2635.440230	37.571767	0.822389
XGBRegressor + ExtraTreesRegressor	2675.499324	38.001762	0.819689
XGBRegressor + ElasticNet + ExtraTreesRegressor	3015.342956	41.926559	0.796786
ElasticNet + ExtraTreesRegressor	3174.633467	43.674026	0.786051
LGBMRegressor + ExtraTreesRegressor	2714.476505	38.441511	0.817063
XGBRegressor + LGBMRegressor + ExtraTreesRegressor	2714.733246	38.420566	0.817045
XGBRegressor + Ridge Regression + ExtraTreesRegressor	2919.321440	41.040093	0.803257
RandomForestRegressor + ElasticNet + ExtraTreesRegressor	2967.801050	42.092253	0.799990
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor	3231.249149	44.810593	0.782236
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor (Method of Stacking)	2777.932698	37.867669	0.812786
AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor	3416.590703	47.534109	0.769745
RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Method of Stacking + Averaging)	2648.322977	37.496234	0.821521
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Method of Averaging)	3388.955730	47.342181	0.771607
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Method of Weighted Average)	3046.770067	43.963393	0.794668
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Method of Stacking)	2804.351333	38.201361	0.811006

TABLE IX: Without implementation of Principal Component Analysis but with implementation of **Adjusted** Auto-Regressive Integrated Moving Average

Models	**MSE**	**MAE**	**R^2**
RandomForestRegressor + ExtraTreesRegressor	2200.807189	33.419437	0.851680
RandomForestRegressor + XGBRegressor + ExtraTreesRegressor	2318.378183	34.814243	0.843757
XGBRegressor + ExtraTreesRegressor	2375.125664	35.529874	0.839933
XGBRegressor + ElasticNet + ExtraTreesRegressor	2779.230866	40.050035	0.812699
ElasticNet + ExtraTreesRegressor	2971.915141	42.016245	0.799713
LGBMRegressor + ExtraTreesRegressor	2391.866946	35.820876	0.838804
XGBRegressor + LGBMRegressor + ExtraTreesRegressor	2357.388640	35.513577	0.841128
XGBRegressor + Ridge Regression + ExtraTreesRegressor	2627.765779	38.859921	0.822906
RandomForestRegressor + ElasticNet + ExtraTreesRegressor	2624.827466	38.644052	0.823104
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor	2860.570593	41.731491	0.807217
DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor (Stacking)	2346.254384	34.085460	0.841878
AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor	2996.580045	44.247916	0.798051
RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor (Stacking + Averaging)	2249.789986	34.135152	0.848379
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Averaging)	2997.674344	44.169096	0.797977
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Weighted Average)	2685.230444	40.900452	0.819034
RandomForestRegressor + AdaBoostRegressor + ExtraTreesRegressor (Stacking)	2430.450019	34.808539	0.836204

RandomForestRegressor + XGBRegressor + ExtraTreesRegressor model is named as the **Model3 Hybrid Estimator**.

Correspondingly, in Table VII, we refer to the Random Forest + ExtraTreesRegressor as the **Model1 Hybrid Estimator (Model1)**, the Random Forest + AdaBoostRegressor + XGBoost + ExtraTreesRegressor as the **Model2 Hybrid Estimator (Model2)**, and the Decision Tree + AdaBoostRegressor + Random Forest + ExtraTreesRegressor as the **Model4 Hybrid Estimator (Model4)**.

Furthermore, as seen from Tables VIII and IX, the chosen models reflect a significant performance improvement in the metrics when utilizing Adjusted ARIMA. This further validates our choice since the best-performing models under the Base ARIMA category also remain the best performers in the Adjusted ARIMA environment. The decisions thus made in the development of these hybrid estimators are therefore valid and justified. We shall proceed to the discussions and conclusion sections of this paper, where our primary focus shall be Tables VI and VII. It is to be noted that the discussion on the above mentioned tables can also be applied to Tables VIII and IX all the models show minor improvement in the performance metrics of the adjusted ARIMA category.

IV. Results and Discussion

The discussion analyzes the performance of hybrid regression models with and without Principal Component Analysis (PCA) using MSE, MAE, and R^2. Results indicate that PCA generally reduces model performance by removing important features, as seen in the **Model1** Hybrid Estimator, where MSE increases and R^2 drops with PCA. Hybrid models combining multiple algorithms perform better, with the **Model2** Hybrid Estimator (RandomForestRegressor + AdaBoostRegressor + XGBRegressor + ExtraTreesRegressor) achieving strong results without PCA. ExtraTreesRegressor plays a key role in improving stability and accuracy by reducing variance and enhancing generalization. Ensemble techniques such as stacking and averaging further enhance performance. **Model2** (Stacking + Averaging) and **Model4** (DecisionTreeRegressor + AdaBoostRegressor + RandomForestRegressor + ExtraTreesRegressor) perform well, proving the effectiveness of combining multiple learning strategies. While PCA-based models help in dimensionality reduction, models without PCA, such as **Model1** and **Model2**, offer higher predictive accuracy.

The paper examines the use of scatter plots to demonstrate the performance of the two proposed models. Each scatter plot illustrates the relationship between the actual AQI and the predicted (averaged) AQI values for the best configuration of each model in different experimental setups.

The overall trend across all figures indicates that both models are capable of capturing the general pattern in AQI values, as evidenced by the strong positive correlation and the clustering of data points around the ideal diagonal (where predicted equals actual). However, **Model1** consistently demonstrates superior performance across all configurations when compared to **Model2**, but this difference in performance is seen only by a very tiny margin (refer Tables VI VII VIII IX)

The scatter plots in Figures 1-4 illustrate the comparison between predicted AQI values (on the y-axis) and actual AQI values (on the x-axis) for **Model1** presented in Tables VI, VII, VIII, and IX. Continuing, the scatter plots in Figures 5-8 illustrate the comparison between predicted AQI values (on the y-axis) and actual AQI values (on the x-axis) for **Model2** presented in Tables VI, VII, VIII, and IX. Each purple dot represents a data point, corresponding to an individual prediction. The red dashed line represents the ideal y = x line, where perfect predictions would lie. All plots above show that, even though some outliers exist, most predictions fall in the direction of the ideal line where the perfect predictions would lie.

Although both models show good promise for real-world AQI forecasting, differences in forecast performance are reasonably small (refer Tables VI VII VIII IX). **Model1**, with a hybrid consisting of only two models, has a simpler architecture and good accuracy, indicating that it is a good and efficient choice. **Model2**, combining four different models, has a better generalization on diverse conditions—most probably due to higher model diversity and ensemble depth.

With this compromise between model complexity and generalization power, the decision between the two can then be based on application-specific needs like interpretability, computational feasibility, and deployment requirements, considering the size of data as well on which they shall be trained. Future research can be directed towards tuning these ensemble settings or investigating more compact combinations that preserve generalization power at the cost of less computational overhead.

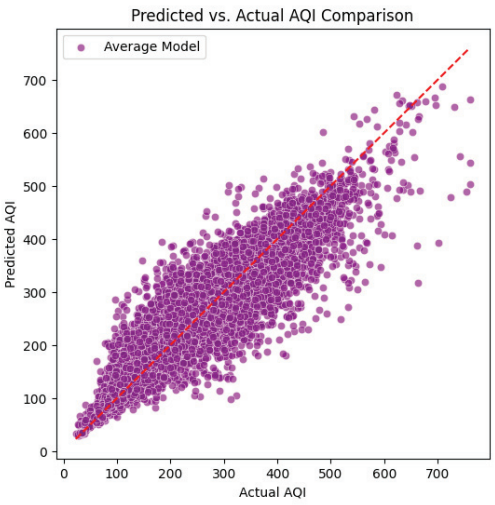

Fig. 1: Predicted (Averaged) vs. Actual AQI values for the best model - **Model1** (Table VI).

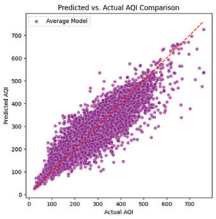

Fig. 2: Predicted (Averaged) vs. Actual AQI values for the best model - **Model1** (Table VII).

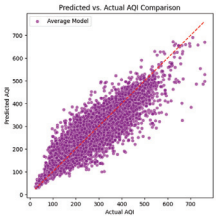

Fig. 3: Predicted (Averaged) vs. Actual AQI values for the best model - **Model1** (Table VIII).

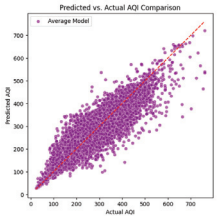

Fig. 4: Predicted (Averaged) vs. Actual AQI values for the best model - **Model1** (Table IX).

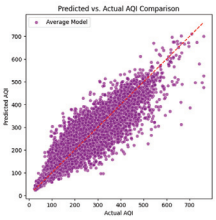

Fig. 5: Predicted (Averaged) vs. Actual AQI values for the best model - **Model2** (Table VI).

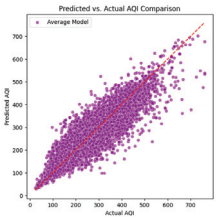

Fig. 6: Predicted (Averaged) vs. Actual AQI values for the best model - **Model2** (Table VII).

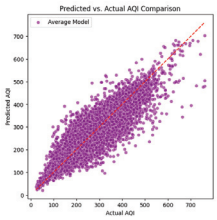

Fig. 7: Predicted (Averaged) vs. Actual AQI values for the best model - **Model2** (Table VIII).

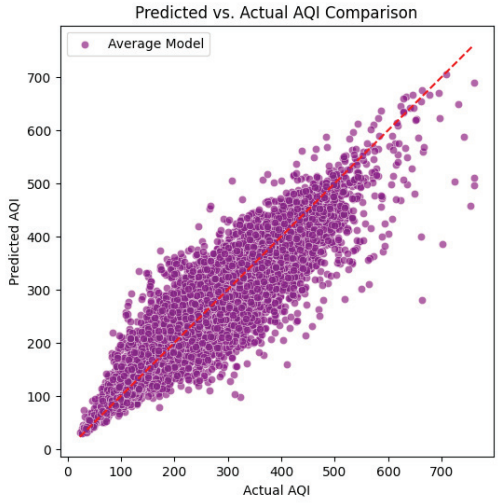

Fig. 8: Predicted (Averaged) vs. Actual AQI values for the best model - **Model2** (Table IX).

V. Conclusion

The **Model2** Hybrid Estimator, which combines RandomForestRegressor, AdaBoostRegressor, XGBRegressor, and ExtraTreesRegressor using stacking and averaging, proves to be the strongest and most generalizable model for regression tasks, especially in noisy, complicated domains such as air quality forecasting. Though easier models like **Model1** (RandomForest + ExtraTrees) have slightly better metrics (e.g., MSE 2200.80 vs. 2249.78), the hybrid's algorithmic synergy of bagging's variance reduction, boosting's error rectification, and gradient boosting's non-linear optimization makes promising results. By exploiting raw feature spaces, Model2 preserves important patterns (e.g., temporal relationships, pollutant interactions) that PCA inadvertently eliminates, as seen from its 15.048% MSE deterioration under PCA versus 13.544% for **Model1**, both under the Adjusted ARIMA category. The meta-learning architecture of the model utilizes heterogeneous, fairly uncorrelated predictions to counteract overfitting threats posed by single-strategy ensembles, including ensembles of gradient boosting models (e.g., XGBoost + LightGBM) or hybrid models of gradient boosting and linear regression (e.g., XGBoost + ElasticNet). These methods show degradation in MSE between 2.44-12.171% under Adjusted ARIMA + PCA increasing up to 22.486%, and 4.56-19.049% under just Adjusted ARIMA increasing up to 24.948% when compared to **Model2**. In addition, its stable MAE (34.135) and competitive R^2 (0.848379) indicate robustness against outliers and stable explanatory strength, essential in high-stakes use cases. In real-world deployment, the capacity of **Model2** to consolidate heterogeneous learning models makes it an excellent candidate for environmental monitoring, financial prediction, or healthcare data analysis, where precision, robustness, and interpretability matter most.

References

[1] Air quality data in India (2015 - 2020). (2020, July 28). Kaggle. https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india

[2] Shikha Sharma. What is Air Quality Index (AQI) & How Is It Calculated? — pranaair.com. https://www.pranaair.com/in/blog/what-is-air-quality-index-aqi-and-its-calculation/, [Accessed 17-12-2024]

[3] National Air Quality Index (AQI) launched by the Environment Minister AQI is a huge initiative under ‘Swachh Bharat ’ — pib.gov.in. https://pib.gov.in/newsite/printrelease.aspx?relid=110654, [Accessed 17-02-2025]

[4] Gupta, N. Srinivasa, Yashvi Mohta, Khyati Heda, Raahil Armaan, B. Valarmathi, and G. Arulkumaran. "Prediction of air quality index using machine learning techniques: a comparative analysis." Journal of Environmental and Public Health 2023, no. 1 (2023): 4916267.

[5] Natarajan, Suresh Kumar, Prakash Shanmurthy, Daniel Arockiam, Balamurugan Balusamy, and Shitharth Selvarajan. "Optimized machine learning model for air quality index prediction in major cities in India." Scientific Reports 14, no. 1 (2024): 6795.

[6] Aram, S. A., 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 21, no. 2 (2024): 1345-1360.

[7] Madan, Tanisha, Shrddha Sagar, and Deepali Virmani. "Air quality prediction using machine learning algorithms–a review." In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 140-145. IEEE, 2020.

[8] Kim, Donghyun, Heechan Han, Wonjoon Wang, Yujin Kang, Hoyong Lee, and Hung Soo Kim. "Application of deep learning models and network method for comprehensive air-quality index prediction." Applied Sciences 12, no. 13 (2022): 6699.

[9] Almaliki, Abdulrazak H., Abdessamed Derdour, and Enas Ali. "Air Quality Index (AQI) Prediction in Holy Makkah Based on Machine Learning Methods." Sustainability 15, no. 17 (2023): 13168.

[10] Ameer, Saba, Munam Ali Shah, Abid Khan, Houbing Song, Carsten Maple, Saif Ul Islam, and Muhammad Nabeel Asghar. "Comparative analysis of machine learning techniques for predicting air quality in smart cities." IEEE access 7 (2019): 128325-128338.

[11] Dragomir, Elia Georgiana. "Air quality index prediction using K-nearest neighbor technique." Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics, LXII 1, no. 2010 (2010): 103-108.