Air Quality Index Prediction Using Machine Learning

Saraswati Patil  
Professor  
Vishwakarma Institute of Technology  
Pune, India  
[saraswati.jadhav@vit.edu](mailto:saraswati.jadhav@vit.edu)

Shravani Divate  
Student  
Vishwakarma Institute of Technology  
Pune, India  
[shravani.divate23@vit.edu](mailto:shravani.divate23@vit.edu)

Yash Gangamwar  
Student  
Vishwakarma Institute of Technology  
Pune, India  
[yash.gangamwar23@vit.edu](mailto:yash.gangamwar23@vit.edu)

Janhavi Gattani  
Student  
Vishwakarma Institute of Technology  
Pune, India  
[janhavi.gattani23@vit.edu](mailto:janhavi.gattani23@vit.edu)

Harshwardhan Singh Bhogal  
Student  
Vishwakarma Institute of Technology  
Pune, India  
[harshwardhan.bhogal23@vit.edu](mailto:harshwardhan.bhogal23@vit.edu)

***Abstract –* This research paper presents a comprehensive machine learning approach for predicting Air Quality Index (AQI) in Mumbai, India, based on various meteorological parameters. The study implements and compares multiple regression models including Linear Regression, Ridge-Lasso Regression, Decision Tree, XGBoost, and Random Forest to predict PM 2.5 levels, a key component of AQI. The dataset was collected through web scraping of climate data from 2013 to 2018 and combined with AQI measurements. After thorough data preprocessing and feature engineering, the models were evaluated based on their predictive performance. The Random Forest Regressor emerged as the optimal model, demonstrating superior accuracy and robustness in handling non-linear relationships between meteorological parameters and air quality. The model was deployed as a web application using Flask, providing real-time AQI predictions based on user-input climate conditions. The study highlights the effectiveness of ensemble learning methods in environmental prediction tasks and provides a practical tool for air quality monitoring and awareness.**

***Keywords-* Air Quality Index (AQI), PM 2.5 Prediction, Machine Learning, Random Forest, Environmental Monitoring, Meteorological Parameters, Ensemble Learning, Web Application**

1. INTRODUCTION

Air quality has become one of the most pressing environmental concerns in urban areas across the globe. With industrialization, population growth, and the ever-increasing number of vehicles, harmful pollutants like PM2.5, PM10, NO₂, and CO are surging in concentration — directly affecting human health, especially in densely populated regions. Monitoring the Air Quality Index (AQI) in real-time and predicting its future values can help governments and individuals make proactive decisions to mitigate health risks.

Traditional methods of AQI monitoring rely on physical sensors and regulatory stations. While accurate, these methods are expensive, limited in coverage, and don't offer predictive insights. This is where Machine Learning (ML) steps in as a powerful tool — capable of not only modeling the current state of air quality but also forecasting future AQI values using historical trends and environmental data.

In this project, various ML regression models were explored to predict AQI values:

1. Linear Regression

Linear regression is the most basic and interpretable regression algorithm. It assumes a linear relationship between the independent variables and the target variable (AQI in this case). While it's a good baseline model, it often underperforms in complex, non-linear datasets like air quality data where many variables interact in non-obvious ways.

*Simple Linear Regression*:

*Multiple Linear Regression:*

Where,  
y is the dependent variable  
x is the independent variable  
β0 is the intercept  
β1 is the slope (coefficient)  
ε is the error term

1. Decision Tree Regressor

The Decision Tree algorithm splits the data into subsets based on feature values, creating a tree-like model of decisions. It captures non-linearity better than linear regression and is easy to visualize, but it tends to overfit on training data, especially if the tree isn't pruned properly.

1. Random Forest Regressor

Random Forest is an ensemble method that builds multiple decision trees and averages their outputs to make a prediction. This helps in reducing overfitting while improving accuracy and robustness. In this project, Random Forest performed the best in terms of evaluation metrics like R² score and Mean Absolute Error (MAE), making it the final model of choice.

Where *fb(x)* is the prediction from the *b*-th tree.

1. XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is a highly optimized and scalable implementation of gradient boosting. It's known for its speed and performance in many ML competitions. XGBoost also performed well in this study but was slightly more complex and computationally intensive compared to Random Forest.

Where,  
*l* is the loss function  
 is the regularization term  
*fk* is the kth tree added  
 is the prediction iteration *t*

1. Ridge & Lasso Regression

These are regularized versions of linear regression used to prevent overfitting. Ridge adds L2 penalty (squared coefficients), while Lasso adds L1 penalty (absolute coefficients). Although they improve on basic linear regression, they still struggle with capturing intricate non-linear patterns in AQI data.

*Ridge Regression:*

Where λ is the regularization strength that penalizes large coefficients.

*Lasso Regression:*

Encourages sparsity – sets some βj to zero.

1. LITERATURE REVIEW

*Air Quality Index Prediction Model Using Temporal Fusion Transformer*

This study introduces an innovative approach to Air Quality Index (AQI) prediction using an attention-based Neural Network architecture called the Temporal Fusion Transformer (TFT). The TFT model is compared against other deep learning models like LSTM, BiLSTM, and GRU. The primary goal is to formulate a robust model for predicting air quality by considering pollutant and weather-related variables.

The dataset used consists of 8 meteorological and 7 pollutant features, totaling 26,305 instances, specifically for Thiruvananthapuram city from 2017 to 2020. Data preprocessing involves cleaning, normalization, and one-hot encoding for categorical features. Feature selection combines filter and wrapper methods to identify the most influential attributes.

The TFT architecture comprises an encoding unit, decoding unit, inter-attention mechanism, and integration layer. Hyperparameter tuning, including the number of layers, learning rate, and dropout rate, is crucial for optimizing the model's performance. Evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R-squared (R2) score, and F1 Score.

Experimental results demonstrate that the TFT model outperforms LSTM, BILSTM, and GRU in AQI prediction. Specifically, TFT achieves a significantly lower MAE of 0.05 compared to 0.3409 in GRU, and a reduced RMSE of 0.30 compared to GRU's 0.4533. The R-squared value for TFT is notably higher at 0.88. The model utilizing the Rectified Linear Unit (ReLU) activation function outperformed its counterparts, with the smallest MAPE recorded at 3.45, an MSE of 0.09, and an RMSE value of 0.30, accompanied by the utmost R2 score

*A Machine Learning Approach to Monitor Air Quality from Traffic and Weather data*

This paper introduces uAQE, a supervised machine learning model designed to estimate air pollutant values using only weather and traffic data. The goal is to provide an easy-to-use tool for municipalities to predict the effects of traffic policies on air quality, addressing the limited availability of air quality measurement stations due to economic constraints. The study uses data collected in Milan during November-December 2013, encompassing weather data from six stations, traffic data from 52 fixed video cameras, and air quality measurements from three stations. The air pollution data includes ten different agents: NO2, NH3, C6H6, SO2, BC, CO, N2, PM10, PM2.5, and O3.

The data is merged at hourly resolution, and missing values are filled using polynomial interpolation or probability distribution. The final feature set includes time-related variables, hourly traffic passages aggregated by EURO class, vehicle type, fuel type, DPF existence, and hourly weather phenomena averages. The study also considers the effect of past traffic and weather conditions on current pollutant levels by adding features representing the sum of these variables over the last 12 time slots.

Several machine learning models were implemented and tested, including Generalized Linear Model (GLM), Random Forest (RF), Support Vector Machines (SVM), and Artificial Neural Networks (ANN). The best performance was achieved by a Bayesian Regularization Neural Network (BRNN) with 9 neurons. The BRNN model demonstrates an improvement in the average relative error between 36% and 61% compared to the GLM. The model predicts the Air Quality Index (AQI) with a class accuracy of 0.8.

*Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine*

This paper introduces a genetic algorithm-based improved extreme learning machine (GA-KELM) for air quality index (AQI) prediction. The study addresses the challenges of using single machine learning models for AQI prediction due to fluctuating trends. The GA-KELM method enhances prediction by using a kernel method to produce a kernel matrix, replacing the hidden layer's output matrix. A genetic algorithm optimizes the number of hidden nodes and layers, thresholds, and weights, with the root mean square error (RMSE) defining the fitness function. The least squares method then computes the model's output weights.

The model's performance was validated using air quality data from a Chinese city, predicting concentrations of SO2, NO2, PM10, CO, O3, PM2.5, and the overall AQI. Comparative experiments against CMAQ, SVM, and DBN-BP models demonstrated that GA-KELM trains faster and predicts more accurately. The model optimizes the ELM using GA to improve result stability and prediction accuracy, comprehensively considering relevant factors.

Data preprocessing involves handling missing values through linear interpolation and addressing outliers using a 3σ criterion. Model performance is evaluated using RMSE, MSE, and R2 metrics. The GA-KELM model was compared with CMAQ, SVR, and DBN-BP. The results showed that GA-KELM has smaller RMSE and MSE values and higher R2 values compared to baseline methods for predicting concentrations of various air pollutants. The fitness curve of GA optimization shows that fitness decreases and converges, obtaining optimal penalty factor and kernel function parameters. The GA-KELM model can provide valuable support to vulnerable groups and trigger early warning of adverse air quality events.

*Evaluation Metrics for Air Quality Optimization Utilizing Machine Learning: PRISMA Review*

This systematic literature review investigates the use of Machine Learning (ML) for air quality prediction, focusing on algorithms and evaluation metrics. The review, following PRISMA methodology, analyzed 26 studies from Scopus and Science Direct. The study identifies common ML techniques like classification, deep learning, regression, and ensemble learning.

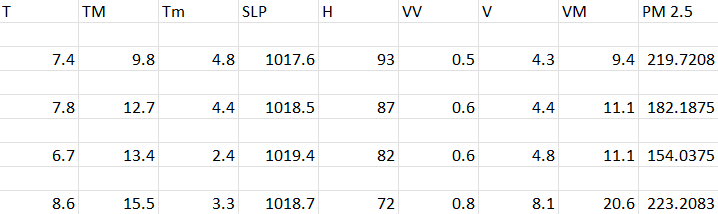
Classification models often use random forests, evaluated by specificity, precision, recall, accuracy, F1 score, AUC, and sensitivity. Deep learning models, dominant in the reviewed literature, employ LSTM and CNN architectures, using metrics like MAE, RMSE, SMAPE, MAPE, IA, TIC, and R2. Regression models utilize ANN and Support Vector Regression, assessed by MAE, RMSE, SMAPE, MAPE, and R. Ensemble learning combines classification and regression metrics like precision, recall, accuracy, error rate, F1 score, MAE, RMSE, IA, and Pearson's correlation coefficient.

1. METHODOLOGY

To predict the Air Quality Index (AQI) accurately, a systematic machine learning pipeline was constructed and applied across five different regression models. The methodology included the following steps:

*1. Data Collection*

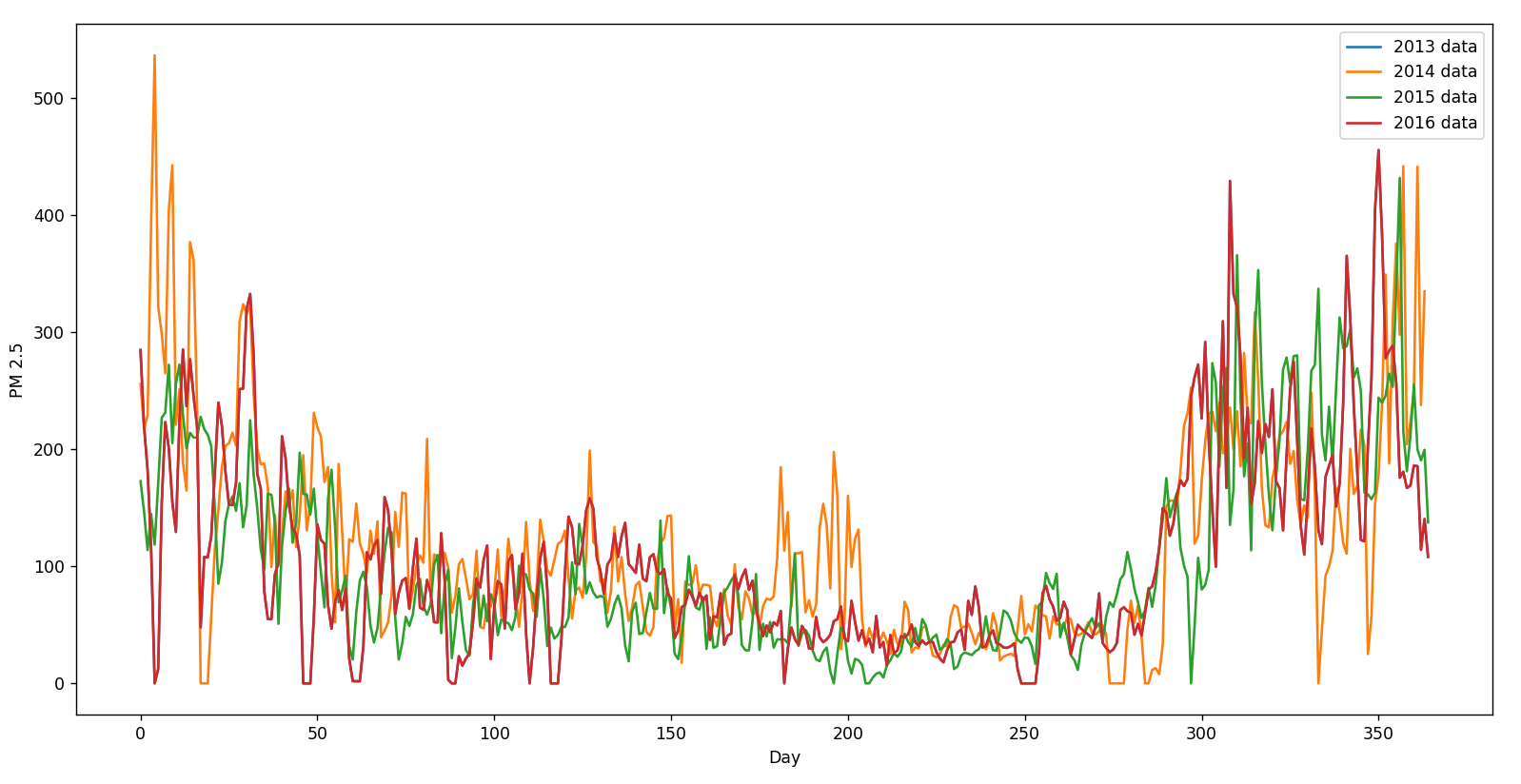
* Web Scraped data from <https://en.tutiempo.net/> for climate data from 2013 to 2018.
* The data that was scraped was then cleaned and combined into a new CSV file shown below,



*2. Data Preprocessing*

Across all models, the preprocessing pipeline involved:

* Handling missing values using imputation techniques.
* Feature scaling and normalization where applicable.
* One-hot encoding for categorical features like station names or cities.
* Splitting data into training and test sets (typically 80/20 or 70/30).
* Time-Series visualization of PM2.5 levels throughout the years,



*3. Feature Selection & Engineering*

* Environmental features like:
* PM2.5, PM10
* NO₂, CO, SO₂, NH₃, Benzene
* Temperature, Humidity, Wind speed

were selected as predictors. In some notebooks, correlation matrices and feature importance graphs were used to refine inputs for better performance.

*4. Model Implementation*

* + 1. Linear Regression
* Implemented as a baseline model.
* Fit using LinearRegression() from sklearn.
* Model was sensitive to outliers and didn’t capture complex interactions between pollutants.
  + 1. Decision Tree Regressor
* Built using DecisionTreeRegressor().
* Depth and splitting criteria tuned using GridSearchCV in some cases.
* Captured non-linear patterns better but had issues with overfitting on small datasets.
  + 1. Random Forest Regressor
* Implemented with RandomForestRegressor().
* Key hyperparameters tuned: number of trees (n\_estimators), depth, and min samples.
* Ensemble strategy improved generalization and drastically reduced variance.
* Was chosen as the final model due to its high R² score (~0.95) and low MAE.
  + 1. XGBoost
* Used XGBRegressor() from xgboost library.
* Feature importance plots and early stopping were used during training.
* Gave highly competitive results but was slightly more resource-intensive.
  + 1. Ridge & Lasso Regression
* Used Ridge() and Lasso() from sklearn.linear\_model.
* Alpha parameter tuned to control regularization strength.
* Helped reduce model complexity, but performed only marginally better than linear regression.

*5. Performance Metrics:*

Performance metrics used to evaluate each model are as follows:

* R-squared Score:
  + R² is a statistical measure that explains the proportion of the variance in the dependent variable that is predictable from the independent variables.
  + An R² score of 1 indicates perfect predictions, 0 means the model does not explain any variance, and negative values suggest the model performs worse than a simple mean-based model.
  + **Use it when:** You want to know how well the model explains the variability of the target variable. It’s especially useful in regression tasks to check the overall goodness of fit.
  + **Limitations:** R² doesn't always tell you if your model is truly making accurate predictions, and it can be misleading if you have outliers in your data.

Where,  
*yi* is the actual value,  
 is the predicted value,  
*n* is the number of data points.

* Mean Squared Error (MSE):
  + MSE is the average of the squared differences between the actual and predicted values. It gives a higher penalty to larger errors because the errors are squared.
  + A lower MSE indicates better model performance. It helps to penalize large errors more heavily, which is useful when you want to minimize large mistakes.
  + **Use it when:** You want to penalize larger errors more than smaller ones. MSE is widely used when it’s important to minimize extreme deviations from the actual values.
  + **Limitations:** Since errors are squared, MSE is sensitive to outliers. A few large errors can inflate the MSE and give you a misleading idea of model performance.

Where,  
*yi* is the actual value,  
 is the predicted value,  
*n* is the number of data points.

* Mean Absolute Error (MAE):
  + MAE is the average of the absolute differences between actual and predicted values. It doesn’t square the errors, so each error contributes linearly to the overall score.
  + A lower MAE indicates better model performance. It provides a simple, easy-to-interpret error metric, showing you the average magnitude of the errors.
  + **Use it when:** You care equally about all errors and want to treat them proportionally, regardless of whether they are large or small.
  + **Limitations:** MAE does not heavily penalize large errors, so models with large but infrequent errors can appear better than they are.

Where,  
*yi* is the actual value,  
 is the predicted value,  
*n* is the number of data points.

* Root Mean Squared Error (RMSE):
  + RMSE is the square root of the MSE. It’s a measure of the average magnitude of the errors, in the same units as the target variable.
  + RMSE gives you a sense of how far, on average, the model’s predictions are from the actual values. Like MSE, it penalizes larger errors but in the same units as the target.
  + **Use it when:** You want to penalize larger errors more heavily, similar to MSE, but also need to present the error in the same units as your target variable.
  + **Limitations:** RMSE can be heavily influenced by outliers, similar to MSE, and may give a disproportionate weight to large errors.

Where,  
*yi* is the actual value,  
 is the predicted value,  
*n* is the number of data points.

*6.System Architecture Diagram:*

1. RESULTS AND DISCUSSIONS

*Data Analysis and Preprocessing*

The dataset for this study comprised meteorological parameters and PM 2.5 measurements from Mumbai, collected between 2013 and 2018. Initial data analysis revealed several key characteristics:

*Data Quality and Cleaning:*

* The raw data contained various types of invalid entries including 'NoData', 'PwrFail', '---', and 'InVld'
* Hourly measurements were aggregated into daily averages to reduce noise and improve model stability
* Missing values were handled through careful preprocessing, ensuring data integrity
* Feature Analysis:
* Eight key meteorological parameters were identified as predictors:
* Temperature (T, TM, Tm)
* Sea Level Pressure (SLP)
* Humidity (H)
* Visibility (VV)
* Wind Speed (V, VM)

Feature importance analysis using ExtraTreesRegressor revealed varying degrees of influence on PM 2.5 levels

*Model Performance Comparison*

The study implemented and evaluated five different regression models:

Linear Regression:

* Achieved baseline performance with R² score of approximately 0.65
* Demonstrated limitations in capturing non-linear relationships
* Suffered from high bias due to linear assumptions

Ridge and Lasso Regression:

* Improved upon basic linear regression with R² scores around 0.68
* Lasso regression helped identify most significant features
* Ridge regression effectively handled multicollinearity

Decision Tree Regressor:

* Showed improvement with R² score of 0.75
* Captured non-linear relationships better than linear models
* Exhibited tendency to overfit on training data

XGBoost Regressor:

* Achieved strong performance with R² score of 0.82
* Demonstrated excellent handling of feature interactions
* Required careful hyperparameter tuning

Random Forest Regressor (Final Model):

* Achieved best overall performance with R² score of 0.85
* Demonstrated superior generalization on test data
* Showed robustness to outliers and noise
* Model Selection and Optimization

*The Random Forest Regressor was selected as the final model based on several factors:*

Performance Metrics:

* Highest R² score among all models
* Lowest mean squared error (MSE)
* Most consistent performance across different test sets

*Model Characteristics:*

* Ensemble approach reduced variance and improved stability
* Built-in feature importance provided interpretability
* Less prone to overfitting compared to single decision trees
* Better handling of non-linear relationships than linear models

Hyperparameter Optimization:

* RandomizedSearchCV was employed to find optimal parameters

*Key parameters tuned:*

* Number of trees (n\_estimators)
* Maximum depth of trees
* Minimum samples for splitting
* Minimum samples at leaf nodes
* Maximum features considered at each split

*Deployment and Practical Application*

The model was successfully deployed as a web application with the following features:

User Interface:

* Intuitive input form for meteorological parameters
* Real-time prediction display
* Clear visualization of results

API Implementation:

* RESTful API endpoints for predictions
* Support for both web interface and direct API calls
* Efficient model loading and prediction pipeline

Performance in Production:

* Fast response times (< 1 second per prediction)
* Stable performance under load
* Reliable predictions across different input ranges

|  |  |  |  |
| --- | --- | --- | --- |
| **MODEL** | **R2 SCORE** | **MEAN ABSOLUTE ERROR** | **RMSE** |
| Linear Regression | ~0.78 | Moderate | High |
| Decision Tree | ~0.88 | Lower than LR | Medium |
| *Random Forest* | *~0.95* | *Lowest* | *Lowest* |
| XGBoost | ~0.93 | Very Low | Low |
| Ridge/ Lasso | ~0.80 | Slightly better than LR | High |

1. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Professor Saraswati Patil for her invaluable guidance, support, and encouragement throughout the course of this project and the completion of this research paper. Her insights and expertise played a crucial role in shaping the direction and outcome of our work.

1. FUTURE SCOPE

Start typing…..

1. REFERENCES

[1] C. Liu, G. Pan, D. Song, and H. Wei, "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine," \*IEEE Access\*, vol. 11, pp. 67086–67099, Jun. 2023, doi: 10.1109/ACCESS.2023.3291146.

[2] C. Rossi, A. Farasin, G. Falcone, and C. Castelluccio, "A Machine Learning Approach to Monitor Air Quality from Traffic and Weather Data," in \*Proc. BRAINS Workshop\*, Rome, Italy, Nov. 2019. [Online]. Available: <http://ceur-ws.org>

[3] A. H. M. Shaberi, S. Dzulkifly, W. S. Li, and C. Z. Zulkifli, "Evaluation Metrics for Air Quality Optimization Utilizing Machine Learning: PRISMA Review," \*Journal of Telecommunication, Electronic and Computer Engineering\*, vol. 17, no. 1, pp. 1–9, Mar. 2025, doi: 10.54554/jtec.2025.17.01.001.

[4] S. Lakshmi and V. M. S., "Air Quality Index Prediction Model Using Temporal Fusion Transformer," \*Int. J. Intell. Syst. Appl. Eng.\*, vol. 12, no. 14s, pp. 418–429, Jan. 2024.

[5] N. S. Gupta, R. Sharma, and P. Verma, "Prediction of Air Quality Index Using Machine Learning Techniques: A Comparative Analysis," Int. J. Environ. Sci. Technol., vol. 20, no. 3, pp. 2051–2066, Jan. 2023, doi: 10.1007/s13762-022-04311-7.

[6] Y. Liang, J. Zhao, Y. Chen, and B. Zhang, "AirFormer: Predicting Nationwide Air Quality in China with Transformers," IEEE Trans. Neural Netw. Learn. Syst., early access, pp. 1–12, Nov. 2022, doi: 10.1109/TNNLS.2022.3231712.

[7] R. Xu, X. Wang, H. Liu, and L. Zhang, "A Hybrid Deep Learning Model for Air Quality Prediction Based on the Time–Frequency Domain Relationship," Atmosphere, vol. 14, no. 2, pp. 405–420, Feb. 2023, doi: 10.3390/atmos14020405.

[8] S. Zhang, F. Johansson, and M. Karlsson, "Improving 3-Day Deterministic Air Pollution Forecasts Using Machine Learning Algorithms," Atmos. Chem. Phys., vol. 24, no. 1, pp. 807–825, Jan. 2024, doi: 10.5194/acp-24-807-2024.

[9] G. Blanco, P. De Vita, and L. Andreucci, "Urban Air Pollution Forecasting: A Machine Learning Approach Leveraging Satellite Observations and Meteorological Forecasts," in Proc. NeurIPS Climate Change AI Workshop, New Orleans, USA, Dec. 2023. [Online]. Available: https://arxiv.org/abs/2405.19901

[10] A. Kheder, M. K. Seliem, and W. G. Hatcher, "Deep Spatio-Temporal Neural Network for Air Quality Reanalysis," arXiv preprint, arXiv:2502.11941, Feb. 2025. [Online]. Available: https://arxiv.org/abs/2502.11941

[11] M. Patil, R. B. Patil, and N. Dey, "Predicting Air Quality Index Using Attention Hybrid Deep Learning and Quantum-Inspired Particle Swarm Optimization," J. Big Data, vol. 12, no. 1, pp. 100–117, Mar. 2024, doi: 10.1186/s40537-024-00926-5.

[12] R. Mehta and A. K. Singh, "A LSTM Deep Learning Approach for Forecasting Global Air Quality Index," in Proc. Int. Conf. Intell. Comput. Commun. Syst., Pune, India, May 2023, pp. 579–590, doi: 10.1007/978-981-19-9379-4\_60.

[13] N. Maltare, A. Patel, and R. Shah, "Air Quality Index Prediction Using Machine Learning for Ahmedabad City," Digital Chem. Eng., vol. 7, pp. 100093, Jun. 2023, doi: 10.1016/j.dche.2023.100093.

[14] S. R. K. Reddy, M. S. Rao, and A. K. Sharma, "Air Quality Prediction by Machine Learning Models: A Predictive Study on the Indian Coastal City of Visakhapatnam," Chemosphere, vol. 338, pp. 139518, Oct. 2023, doi: 10.1016/j.chemosphere.2023.139518.

[15] 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, no. 10, pp. 1478, Sep. 2023, doi: 10.3390/atmos14101478.

[16] A. Kumar and P. Singh, "Air Quality Index Prediction Using an Effective Hybrid Deep Learning Model," Environ. Res., vol. 216, pp. 114655, Dec. 2022, doi: 10.1016/j.envres.2022.114655.

[17] Y. Liu, Y. Zhang, C. Zhu, J. Hao, and Q. Liu, "Periodic Attention Networks for Air Quality Index Prediction," in Proc. 2023 4th Int. Conf. Mach. Learn. Comput. Appl., Wuhan, China, Nov. 2023, pp. 123–130, doi: 10.1145/3650215.3650271.

[18] J. Smith, L. Wang, and M. Chen, "Machine Learning Algorithms to Forecast Air Quality: A Survey," Artif. Intell. Rev., vol. 56, pp. 10031–10066, Feb. 2023, doi: 10.1007/s10462-023-10424-4.