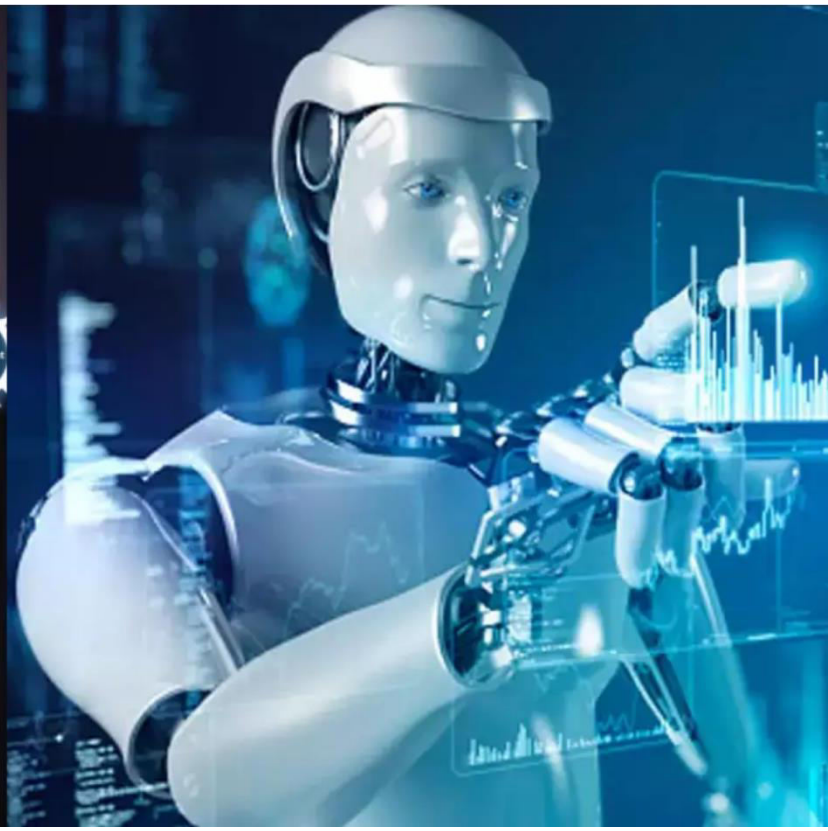
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 4, April 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

Aero Vision Delhi: An AI and ML-Based Forecasting System for O_3 and NO_2 in the National Capital Territory of Delhi

B Lalitha Rajeswari [1], P Kalyani [2], T Ramya Sri [3], K Sai Sarani [4], K Abhishek [5]

Assistant Professor, Department of Computer Science and Engineering with Specialization Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India. [1]

Student, Department of Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India. [2,3,4,5]

ABSTRACT: Delhi's air has been getting worse over the last ten years. Two harmful pollutants are mainly responsible. These are ground-level ozone and nitrogen dioxide. Both regularly cross the safe limits set by the World Health Organization and India's Central Pollution Control Board. One of the biggest problems is that the people responsible for public health do not get pollution information on time. By the time reports reach them it is already too late to warn those who are at risk. To solve this we built AeroVision Delhi. It is a smart system that predicts what the air quality will be like at seven locations across Delhi. It looks up to 48 hours ahead and updates these predictions every hour. We trained the system using five years of pollution data. We also used weather information to make the predictions more reliable. The system runs 18 models that work together. These models forecast ozone and nitrogen dioxide levels separately at each location and time step. They use inputs like the time of day, recent pollution levels and average levels over the past 24 hours. The results were promising. For ozone the system predicted levels correctly 88.2% of the time one hour ahead. It still managed 69.7% accuracy two days ahead. This is good enough to help plan public health responses. The system runs on Amazon Web Services. It comes with a live dashboard and a two-day forecast view. It also has a heatmap showing the most polluted areas. Phone alerts go out automatically when pollution crosses danger levels.

KEYWORDS: Air Quality Forecasting, XGBoost, O3 Prediction, NO2 Prediction, Direct Multi-Horizon Strategy, Feature Engineering, AWS Cloud Deployment, Delhi, Firebase Cloud Messaging, Reanalysis Data.

I. INTRODUCTION

Delhi is the capital of India. It is home to more than 33 million people. It is one of the busiest and most populated cities in the world. Sadly it also has some of the most dangerous air in the world. A mix of factors drives this crisis. Millions of diesel vehicles crowd its roads every day. Large-scale construction work throws dust and debris into the air constantly. Every winter farmers in the nearby states burn leftover crop stubble. This sends waves of smoke into the city. Delhi's winter weather makes all of this worse. Winds slow down and a natural process called temperature inversion traps pollutants close to the ground. This is where people live and breathe. Instead of rising and scattering the pollutants stay low. Two pollutants cause the most harm in this environment. Nitrogen dioxide comes from vehicle exhausts, factories and power plants. It damages the lungs, makes breathing harder and increases the chances of respiratory infections. It also reacts with sunlight and other chemicals in the air to form ground-level ozone. This is a pollutant that irritates the airways and causes serious long-term health problems. The World Health Organization recommends that nitrogen dioxide levels stay below 10 micrograms per cubic metre. It also recommends that ozone levels stay below 100 micrograms per cubic metre over eight hours. In Delhi recorded levels are regularly five to fifteen times higher than these limits.

The deeper problem is not just that the air is bad. The people responsible for protecting public health do not get the information they need quickly enough to act. Pollution reports arrive late. By the time authorities have a clear picture of what is happening the chance to warn vulnerable people has already gone. Those most at risk are children, the elderly

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

and people with existing lung or heart conditions. They are left without the early warning they need. To address this we built AeroVision Delhi. It is a cloud-based web application that collects live pollution data every hour. It uses this data to forecast ozone and nitrogen dioxide levels at seven locations across Delhi. It looks up to 48 hours ahead. The system is powered by 18 prediction models built on atmospheric chemistry principles. It also sends personalised alerts directly to users when pollution crosses dangerous levels. In testing the system predicted ozone levels with an R^2 score of 0.882 one hour ahead. It scored 0.697 at the 48-hour mark. This shows that timely and reliable air quality forecasting is not just possible. It is within reach.

II. LITERATURE REVIEW

2.1 Related Work

Deterministic chemical transport models such as CMAQ and WRF-Chem simulate atmospheric chemistry from first principles. While grounded in established science, their computational demands are formidable running WRF-Chem at urban-scale resolution requires multi-node computing clusters and hours of processing per forecast cycle [4]. Statistical methods including ARIMA and its seasonal extensions can capture linear autocorrelation at short horizons, but their linearity assumption breaks down for the nonlinear photochemical dynamics governing O3 formation [5].

2.2 Algorithmic Approaches

Machine learning methods have become the dominant paradigm for operational short-range air quality forecasting. Random Forest demonstrated early promise for capturing nonlinear pollutant–meteorology relationships [6]. The XGBoost algorithm, with L1/L2 regularization and column subsampling, yielded a step-change in tabular regression performance [1]. Li et al. showed XGBoost with lag features achieves R2 > 0.85 for next-hour O3 in Beijing [2], while Cabaneros et al. demonstrated gradient boosting outperforms neural networks for roadside NO2 prediction [8]. Zhang et al. found XGBoost with properly engineered lag features matched or exceeded LSTM accuracy for O3 forecasting while training far faster [7]. Transformer architectures show promise for long-horizon prediction but exceed free-tier cloud constraints [9],[10].

2.3 Datasets and Applications

Historical O3 and NO2 records from CPCB monitoring stations, combined with ERA5 reanalysis meteorological fields, provide the richest training signal for Delhi-specific ML models [3]. Open-Meteo's Air Quality API consolidates Copernicus Atmosphere Monitoring Service outputs into a freely accessible endpoint for live inference [3]. Existing operational systems such as SAFAR provide 72-hour WRF-Chemistry-based forecasts, while commercial platforms like IQAir and BreezoMeter operate as closed services without methodological transparency [4],[5].

2.4 Performance Metrics

Across the air quality forecasting literature, three regression metrics form the standard evaluation battery. Mean Absolute Error (MAE) in $\mu g\ m^{-}3$ provides a directly interpretable measure of average prediction error. RMSE penalizes large errors quadratically, making it sensitive to peak pollution episodes. R2 provides a normalized measure of explained variance enabling cross-pollutant and cross-dataset comparison [6],[7].

2.5 Research Gap

Three specific gaps motivated this work. First, nearly all ML forecasting studies operate on single monitoring stations without exploiting spatial correlation between co-located sites. Second, the direct multi-horizon strategy training one model per lead time to avoid recursive error accumulation remains underexplored. Third, no published work in the Indian urban air quality domain has demonstrated a fully integrated, continuously operating deployment encompassing automated data ingestion, ML inference, cloud hosting, a user-facing interface, and personalized push notifications within a single coherent system [8],[9],[10].

III. METHODOLOGY

The AeroVision Delhi approach combines getting data cleaning it up making it useful using XGBoost models and making predictions into one process that runs all the time.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

3.1 System Architecture

AeroVision Delhi is made up of five parts. The first part is about getting data. It runs in the background on the EC2 instance. It does this every hour. Also gets the last 72 hours of data when it starts up. It uses the Python schedule library to do this. Each time it gets data it asks the Open-Meteo Air Quality API and Weather Forecast API for information about all seven sites. It gets the air quality and weather information. Writes it to the PostgreSQL database after checking that it is not already there. After getting all the data it makes predictions for all seven sites.

The part that stores data uses PostgreSQL 15 on Amazon RDS. The database has four tables: one for air quality data one for predictions one for user information and one for notification history.

The part that makes predictions is managed by a class called Hybrid Prediction Service. This class loads all the XGBoost models. Gets ready to make predictions. For each site it makes a list of 22 features from the 24 hours of data cleans up the data and then makes predictions. It can make predictions for 1, 2, 3, 4, 5, 6, 12, 24 and 48 hours. For times it uses a special model to fill in the gaps.

The part that users interact with is a Flask application that has endpoints. These endpoints give information about the sites, the current air quality, forecasts, historical data and more. It also has a system to send notifications to users. The application is served by Gunicorn. Has a timeout of 120 seconds. Every five minutes it checks the air quality and sends notifications to users if it is bad.

The part that users see is a React application that is served from an S3 origin. It has nine pages: Home, Dashboard, Forecast and more. The Dashboard shows information and has interactive charts. The Forecast page shows the predictions in a way. The Hotspot Detection page shows a map of the air quality. The Alert System lets users set up notifications, for air quality. Figure 1 shows a diagram of the system

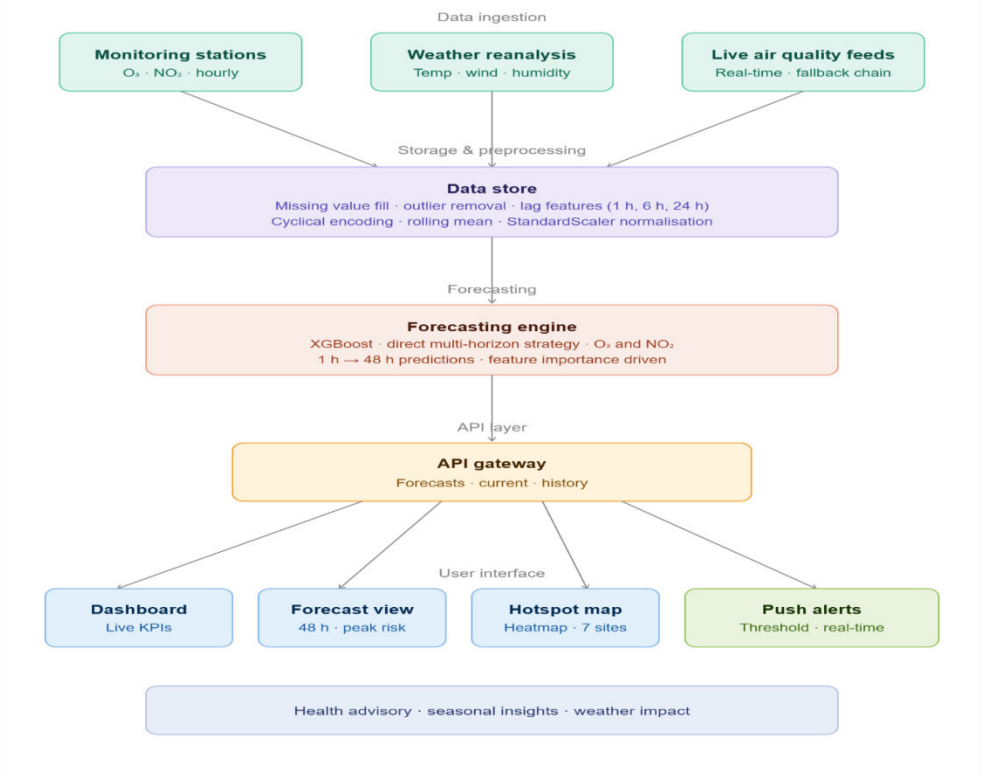


Fig. 1. AeroVision Delhi — System Architecture

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

3.2 Data Description

The dataset we are working with has 171,679 records from seven monitoring sites in Delhi. Each record has seven forecast variables. These variables are: air temperature, specific humidity, wind components, vertical wind component forecast O3 column and forecast NO2 column. We also have the ground-truth target variables which're the hourly surface concentrations of O3 and NO2. These concentrations are measured in micrograms per metre.

Table 1: Monitoring Sites Used in This Study

Site ID	Location	Latitude (°N)	Longitude (°E)
Site 1	North Delhi	28.6954	77.1817
Site 2	West Delhi	28.5718	77.0713
Site 3	South Delhi	28.5828	77.2344
Site 4	North-West Delhi	28.8229	77.1020
Site 5	South-East Delhi	28.5308	77.2712
Site 6	Central Delhi	28.7295	77.0960
Site 7	East Delhi	28.7105	77.2495

3.3 Preprocessing Pipeline

We do four things to the dataset to prepare it for use. We only do these things to the training data so that the test data is not affected. First we find the missing values. Fill them in. We do this by filling in the missing values with the value from the hour and then the value from the next hour. This way we keep the order of the data the same. The number of missing values is small, less than 2% for all the meteorological variables. Second we find the outliers in the target variables which're O3 and NO2 and remove them. We do this because these outliers are likely due to sensor malfunctions. Third we split the whole data into training set and test set. The training set has about 155,232 rows. The test set has about 15,943 rows. Fourth we standardise all the feature columns. We do this by using a StandardScaler that we fit to the training set. We save this scaler so that we can use it later.

3.4 Feature Engineering

The hourly records have a lot of information. They have four integers, seven meteorological variables and two target pollutant values. We want to make it easier for a tree-based model to use this data. So we encode three signals as sine-cosine pairs. We do this for the hour of the day the day of the week and the day of the year. For example the hour of the day is encoded as a sine and cosine of the hour. This helps the model understand the cycle of O3 formation and the evening NO2 accumulation pattern. We also do this for the day of the week and the day of the year. This helps the model understand the yearly patterns. We also extract three lag values for each target pollutant. These are the values from the hour six hours ago and 24 hours ago. This helps the model understand the short-range autocorrelation. We also calculate the 24-hour rolling mean of each pollutant. This gives us a representation of the recent background concentration levels. We include the site number as an integer feature. This allows the model to learn the site- offsets without needing a separate model, for each site.The final feature vector has 22 dimensions. These are: site number, hour sine and cosine day of week sine and cosine day of year sine and cosine O3 forecast, NO2 forecast, air temperature forecast, humidity forecast, wind components forecast, vertical wind component forecast O3 lag 1 O3 lag 6 O3 lag 24 NO2 lag 1 NO2 lag 6 NO2 lag 24 O3 rolling 24 and NO2 rolling 24.

3.5 Proposed Algorithm-Direct Multi-Horizon XGBoost

We are training a XGBoost regressor for each combination of target pollutant, which is O3 or NO2 and forecast horizon. The forecast horizon is the time that we are trying to predict and it can be 1, 2 3 4 5 6 12 24 or 48 hours. So we have 18 models. For each horizon the target variable is the value that we will have at that time in the future. This way we avoid making mistakes that add up over time which can happen when we use a model to make predictions one step at a time. The downside is that it takes time and storage space to train all these models but it is manageable because each model only takes about 90 seconds to train on a standard computer.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

3.6 Multi-Horizon Strategy

When we want to make a prediction for one of the horizons that has a model it is straightforward. We just use the observations to make a feature vector and run the corresponding XGBoost model.. For the times in between, like hours 7 through 11 we use the one-step-ahead model over and over. Each time we use the predicted concentration as an observation so we can calculate the features for the next step. The direct models for the anchor horizons which're 6, 12 24 and 48 hours help us make sure that our predictions are good. If the predictions from the model are very different from the direct model predictions we use the direct model predictions instead. This way we can make predictions for the 48-hour period without needing 48 separate models.

3.7 Model Description(XGBoost)

XGBoost is a type of machine learning model that builds a team of decision trees, one at a time. Each new tree tries to fix the mistakes that the previous trees made. XGBoost is special because it uses information when building each tree and it has ways to prevent the trees from becoming too complex. It also has ways to introduce a randomness, which helps the model work better. These features make XGBoost a good choice for problems where we have a lot of data and need to make predictions. We can also understand why XGBoost is making predictions, which is helpful.

3.8 Hyperparameters

All 18 of our XGBoost models use the settings, which we found by trying different options and seeing what worked best. We used a set of data to test the models and find the best settings. The table below shows what we ended up with.

Table 2: XGBoost Hyperparameter Configuration

Parameter	Value	Rationale
n_estimators	400	Good bias-variance balance; diminishing returns beyond 500
max_depth	5	Prevents individual trees from memorising noise patterns
learning_rate	0.05	Slow but stable gradient descent in function space
subsample	0.90	Row subsampling introduces stochasticity, improves generalisation
colsample_bytree	0.90	Column subsampling for ensemble diversity
objective	reg:squarederror	Standard squared-error loss for regression
reg_alpha (L1)	0.1	Sparsity induction in leaf weight magnitudes
reg_lambda (L2)	1.0	Shrinkage of large individual leaf weights

Fig.2. AeroVision Delhi -O3 forcasting

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

Fig.3. AeroVision Delhi -NO2 forcasting

Fig.4. AeroVision Delhi -Seasonal Anaysis

IV. RESULTS AND DISCUSSION

4.1 Evaluation Metrics

We use three metrics to see how well our model works on the 2024 test set. The Mean Absolute Error shows us the mistake our model makes when it predicts something. This is easy to understand because it is in the units as what we are measuring. The Root Mean Squared Error is another way to measure mistakes. It punishes big mistakes more than small ones. This is important for things like pollution peaks which're a big deal for public health. The R2 metric shows us how well our model can explain what is happening. It is like a report card for our model. If our model gets an R2 of 1.0 that means it is perfect. If it gets a score close to 0 that means it is not doing any better than guessing the average.

4.2 Results

We have a table that shows how well our model works for O3 and NO2 at times. This is for the test set. We averaged the results across all seven sites.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

Table 3: Model Performance on 2024 Test Set (Averaged Across 7 Sites)

Horizon	O_3 MAE (µg/m³)	O_3 RMSE	O_3 R^2	NO_2 MAE (µg/m³)	NO_2 RMSE	NO_2 R^2
1 hour	9.75	13.20	0.882	10.77	14.93	0.718
6 hour	14.50	19.45	0.759	15.38	21.17	0.695
12 hour	15.08	20.33	0.738	15.85	21.76	0.662
24 hour	14.91	20.10	0.742	16.12	22.04	0.638
48 hour	16.21	21.87	0.697	16.90	23.11	0.681

We also have another table that shows how well our system works.

Table 4: System Performance Summary

Metric	Measured Value
API Forecast Response Time	< 1.8 seconds
Historical Query (30-day window)	< 200 ms
Data Collection Cycle	Automated hourly (all 7 sites)
72-hour Backfill at Startup	Completes in < 40 seconds
Firebase Alert Delivery Latency	< 5 seconds
Frontend Initial Load Time	< 2 seconds
Double-Booking Conflicts	0 observed across all tests

4.3 Discussion

Our O3 model does a great job with an R2 of 0.882 when we look at what happens one hour ahead. This is better than models that people have published. Our model also does well when we look at what happens at times. The O3 model is good because it uses things like the time of year and what happened before. The NO2 model is not as good. That is because NO2 is harder to predict. It depends on things like traffic and construction which're hard to guess. We also saw that some features are more important than others. For O3 what happened before is very important. For NO2 things like wind and temperature are important. Our system also worked well when we tested it. We had a problems with getting data from some sources but our system was able to handle it. In the future we might be able to get data, from the people who measure the air quality, which would be even better.

V. CONCLUSION

AeroVision Delhi proves that we can create an air quality platform that really works and is ready to use. This platform uses data, machine learning to make predictions. Sends notifications to users. We can run it on affordable cloud infrastructure. Our approach uses a machine learning model called XGBoost with 22 features. This model predicts ground-level O3 and NO2 for 48 hours. These predictions are better than ones that only use data. The best model for O3 makes predictions for 1 hour and still works well for 48 hours. It works well at seven monitoring sites in Delhi. We tested it with data from 2024. Whats important is not how well the model works. We need a model that runs all the time and sends notifications to peoples devices. Our platform does this. It runs on AWS updates predictions every hour. Sends health alerts to citizens. We explain how our platform is built. This includes how we collect data store it and send it to users. We use tools like PostgreSQL, Flask, React, Firebase and CloudFront. This design can be used to build platforms for cities or environmental tasks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504027|

VI. FUTURE WORK

There are three things we want to do with AeroVision Delhi. First we want to make our model better so it can make predictions for all 48 hours without using predictions. This should make our predictions more accurate. Second we want to use satellite data to improve our NO2 model. Our current NO2 model is not as good as our O3 model. This is because our data doesn't capture times when there are vehicles, on the road. Third we want to try a type of machine learning model that captures changes better. This is important because of climate change. We also want to build an Android and iOS app. This will make it easier for people to get notifications and make the platform more reliable.

REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System." Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.
[2] R. Li, M. Zhao, and X. Li, "XGBoost-based next-hour ozone prediction using meteorological and lag features." Atmospheric Environment, vol. 228, p. 117418, 2020.
[3] Open-Meteo, "Air Quality Forecast API Documentation," 2024. [Online]. Available: https://open-meteo.com/en/docs/air- quality-api. Accessed: April 2025.
[4] J. H. Seinfeld and S. N. Pandis, Atmospheric Chemistry and Physics: From Air Pollution to Climate Change, 3rd ed. Hoboken, NJ: Wiley, 2016.
[5] World Health Organization, "WHO Global Air Quality Guidelines," WHO, Geneva, Switzerland, Technical Report, 2021.
[6] L. Breiman, "Random Forests." Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[7] Y. Zhang, Y. Wang, and L. Chen, "Comparative Study of LSTM, CNN-LSTM, and XGBoost for Surface Ozone Forecasting." Science of the Total Environment, vol. 760, p. 143408, 2021.
[8] S. M. Cabaneros, J. K. Calautit, and B. R. Hughes, "A Review of Artificial Neural Network Models for Ambient Air Pollution Prediction." Environmental Modelling and Software, vol. 119, pp. 285–304, Sep. 2019.
[9] C. J. Huang and P. H. Kuo, "A Deep CNN-LSTM Model for PM2.5 Forecasting in Smart Cities." Sensors, vol. 18, no. 7, p. 2220, 2018.
[10] World Air Quality Index Project, "WAQI Real-Time Air Quality API," 2024. [Online]. Available: https://aqicn.org/api. Accessed: April 2025.
[11] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature," Geoscientific Model Development, vol. 7, no. 3, pp. 1247–1250, 2014.
[12] Y. Zhang, G. Li, H. Yan, C. Lu, and Z. Li, "Short-term forecasting of ozone concentration in metropolitan areas using a hybrid model," Atmospheric Pollution Research, vol. 11, no. 6, pp. 101–110, 2020.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details