URBAN-AIRNet: ROAD-NETWORK-AWARE AI FRAMEWORK FOR URBAN AIR POLLUTION FORECASTING

M. Pravin Kumar, Student, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore

M. Ramaraj, Assistant Professor, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore

Abstract

Urban air pollution is a growing concern in Indian cities like Chennai, where NO_2 and O_3 levels often exceed safe limits. Most forecasting systems treat cities as uniform spaces, missing how road networks shape pollution distribution. URBAN-AIRNet fixes this by combining CPCB air quality data, ERA5 weather data, and OpenStreetMap road network features into one machine learning pipeline. Graph metrics like intersection density, betweenness centrality, and road type score are used alongside weather variables to predict pollution levels. Random Forest outperformed XGBoost with $R^2 = 0.9934$ and RMSE = 2.03 $\mu g/m^3$ for NO_2. The results are shown through a live Streamlit dashboard with maps, forecasts, and explainability charts.

Keywords - *Air pollution, Road network, Machine learning, XGBoost, Random Forest, NO_2, O_3, OSMnx, Feature importance, Streamlit.*

1. INTRODUCTION

Air pollution is one of the biggest health problems in Indian cities today. Cities like Chennai face high levels of NO_2 and O_3 caused mainly by traffic, which leads to breathing problems and long-term health damage. The issue with current systems is that they only predict pollution based on weather, without considering how roads are laid out. A busy intersection on Anna Salai will obviously have more emissions than a quiet residential street, but most models don't account for this.

URBAN-AIRNet is our attempt to bridge that gap. Instead of treating Chennai as one big area with uniform pollution, we used road network properties, specifically how many roads meet at a point, how central that junction is to traffic, and what kind of roads they are, to make predictions more location-specific. The system uses three years of real CPCB data from Chennai, combined with ERA5 weather data and OpenStreetMap road graphs, to train two ML models and display results through an interactive dashboard.The main contributions of this work are: building a road-aware feature pipeline using OSMnx and NetworkX; comparing XGBoost and Random Forest on real Chennai pollution data; measuring both accuracy and training time; and deploying an explainable dashboard on Streamlit Cloud that anyone can access for free.

2. LITERATURE SURVEY

Kumar and Pande [1] worked on predicting AQI across 23 Indian cities using six years of data. They used Random Forest and SVM with feature selection and found that NO_2, CO, and PM2.5 are the strongest predictors. Their preprocessing approach of removing outliers and using median imputation is something we followed closely in our own pipeline. The gap was that their model didn't account for road structure at all.

Natarajan et al. [2] combined Grey Wolf Optimisation with a Decision Tree to predict AQI in six cities including Chennai, reaching 95.22% accuracy there. This gave us a solid benchmark. Their work confirmed that NO_2 and O_3 are the most important pollutants to focus on, which shaped the target variables we chose in URBAN-AIRNet.

Boeing [3] created OSMnx, the Python tool we rely on to download and analyse road networks from OpenStreetMap. His work on computing intersection density, node degree, and betweenness centrality as graph metrics gave us the framework to turn a city's road layout into usable machine learning features.

Petrić et al. [4] showed that using ERA5 reanalysis data as meteorological features significantly improves NO_2 and O_3 predictions, with R^2 gains of up to 26%. This confirmed our decision to include ERA5 weather variables rather than relying solely on the CPCB readings.

Cabaneros et al. [5] specifically studied roadside NO_2 prediction using neural network models. Their key finding was that models which include traffic and road proximity features consistently beat those that only use weather data. This directly supported our approach of adding road network features to the standard meteorological inputs.

Kothandaraman et al. [6] compared six ML models including XGBoost and AdaBoost for PM2.5 prediction using meteorological data. They found XGBoost and AdaBoost to be among the most reliable models in terms of RMSE and MAE. Their comparison methodology inspired the model evaluation approach we used in URBAN-AIRNet.

Gupta et al. [7] predicted AQI across Indian cities using SVR and Random Forest with SMOTE-balanced data, achieving low RMSE values with Random Forest. Their work showed that Random Forest handles correlated pollutant features well, which aligns with our own finding that Random Forest outperformed XGBoost on our dataset.

Shakya et al. [8] worked on PM2.5 prediction in New Delhi using deep learning with meteorological, vehicular, and emission data. Though they focused on a different pollutant and city, their use of multiple data sources together with vehicle-related features is closely related to what we do in URBAN-AIRNet with road graph features.

Mandal et al. [9] proposed a graph neural network approach for simultaneously predicting O_3 and NO_2 in Delhi. Their finding that O_3 is harder to predict than NO_2 due to its secondary formation mechanism matches what we observed in our results, where our O_3 model had noticeably lower R^2 than the NO_2 model.

Malhotra and Aulakh [10] studied the relationship between meteorological factors and air pollutants in Delhi, finding that temperature, wind speed, and humidity significantly affect NO_2 and O_3 concentrations. These findings are reflected in our feature matrix design, where meteorological variables like AT, WS, and RH are included alongside road network features.

Rautela et al. [11] studied the spatiotemporal prediction of aerosol concentrations, emphasising that spatial context matters in pollution modelling. Their work supports the core argument of URBAN-AIRNet: that treating all grid cells in a city the same, without considering their local road characteristics, leads to weaker predictions.

Krishan et al. [12] used LSTM with vehicular emissions and traffic data to forecast O_3, PM2.5, NOx, and CO in Delhi, getting R^2 values between 0.92 and 0.98. While we did not use deep learning, their approach of integrating traffic-related features into the model influenced our decision to include road type score and betweenness centrality as proxy traffic indicators.

Mampitiya et al. [13] compared LightGBM, Random Forest, and other models for PM10 prediction in Sri Lanka and reported that Random Forest had high accuracy with interpretable outputs. Their observation that ensemble models perform well on smaller, correlated datasets applies to our case, where Random Forest achieved R^2 of 0.9934 on our 1,006-record Chennai dataset.

Breiman [14] originally introduced the Random Forest algorithm and established its theoretical properties including resistance to overfitting through bootstrap aggregation and feature randomisation.

Chen and Guestrin [15] introduced XGBoost and demonstrated its superior performance on structured tabular data. Their design principles of regularised boosting and column subsampling are directly reflected in the hyperparameters we used in our XGBoost configuration, and their framework established the benchmark against which Random Forest was compared.

3. DATASET COLLECTION

We used three datasets, all freely available. The main one is CPCB's daily air quality data from the Alandur Bus Depot station in Chennai (2023–2025), downloaded from airquality.cpcb.gov.in. After cleaning, we had 1,006 records with 23 variables covering pollutants like NO_2, O_3, NOx, PM2.5, and weather readings like temperature, humidity, wind speed, and solar radiation. The second source is ERA5 reanalysis from ECMWF, accessed via the Copernicus Climate Data Store. It gives us hourly global weather data resampled to daily resolution. The third is OpenStreetMap road data downloaded using OSMnx for the Chennai area (13.0°N to 13.35°N, 80.1°E to 80.35°E), giving around 50,000 intersections and 100,000 road segments.

4. PROPOSED WORK

URBAN-AIRNet works in five stages: collect data, clean and preprocess, extract road features, train models, and deploy the dashboard. In preprocessing, we standardise column names, handle missing values with median imputation, and remove readings more than three standard deviations away from the mean to filter sensor errors.For road features, OSMnx downloads the Chennai road graph and NetworkX computes four features per grid cell: node degree mean, intersection density, betweenness centrality, and road type score. Only the top 25% most traffic-heavy intersections are kept, filtered by centrality and road class. These road features are merged with the CPCB data to form the final feature matrix used for training.

Two regression models are trained: XGBoost (200 trees, depth 6, learning rate 0.05) and Random Forest (200 trees, depth 10). Both are evaluated on 20% held-out data using RMSE, MAE, R^2, accuracy (within ±10% of actual), and training time. The Streamlit dashboard has five tabs: AQI Map, Forecast, Feature Importance, Model Comparison, and Accuracy & Time, all accessible publicly from Streamlit Cloud.

5. SYSTEM ARCHITECTURE

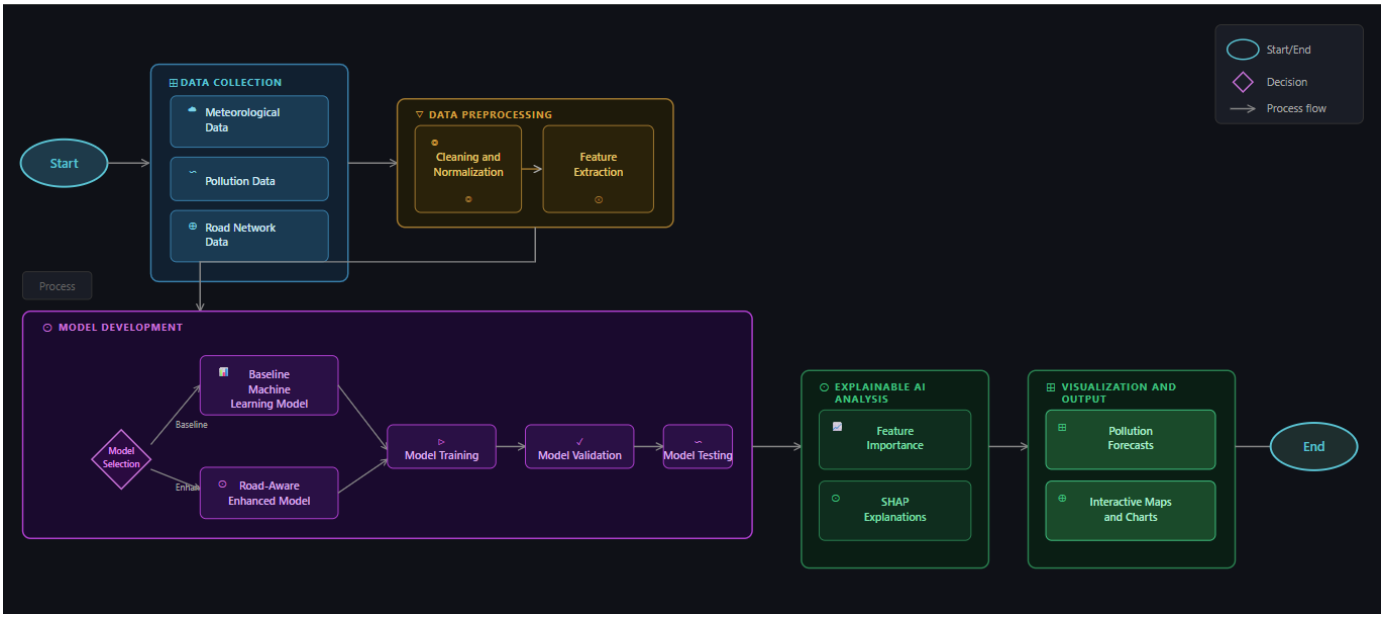

Fig 5.1. Urban-Airnet System Architecture

6. METHODOLOGY

ROAD NETWORK FEATURE EXTRACTION

The Chennai Road network is modelled as a graph G = (V, E), where V is intersections and E is road segments. Betweenness centrality informs us of the number of shortest paths going through each intersection that is an excellent proxy of the traffic flow. The GST Road will automatically be a high-centrality junction with more emissions compared to a low-centrality side road. The degree of a node informs us of the number of roads that were joined at a place. Intersection density is a measure of the density of the road network within an area. Road type score is used to provide a weighted rating of traffic between 1.0 (motorways) and 0.0 (footpaths).

RANDOM FOREST

Random Forest constructs hundreds of decision trees, each of which is trained using a random sample of the data, and a random subset of features. It takes their predictions on average and is therefore much more stable than a single tree and not easily overfit. In our case it achieved $R^2 = 0.9934$, RMSE = 2.03 µg/m³, and 95.3% accuracy for NO_2, training in 12.7 seconds. It also provides us with the scores of the feature importance, informing us of the inputs that actually mattered.

XGBOOST

XGBoost is a tree-building model that uses a successive generation of trees that correct mistakes of the previous ones. It is quick and can work with big data. In our test it achieved $R^2 = 0.9751$, RMSE = 3.95 µg/m³, and 87.5% accuracy, training in only 4.2 seconds. It is three times quicker than the Random Forest but less precise with our data, probably due to the powerful correlation between NOx and NO_2 being more effectively modeled by averaging (Random Forest) than (XGBoost).

EXPLAINABLE AI - FEATURE IMPORTANCE

Feature importance in Random Forest feature importance is used to quantify the reduction of prediction error by each feature in all trees. Our analysis indicated that NOx predominates at 0.9361 which is chemically feasible as NO2 is directly obtained out of NOx. NO follows at 0.0386.

7. RESULTS

Model	RMSE (µg/m³)	MAE (µg/m³)	R^2 Score	Accuracy (%)
XGBoost	3.9517	1.9330	0.9751	87.5%
Random Forest	2.0340	1.3343	0.9934	95.3%

Table 7.1. Model Performance Comparison NO_2 Prediction

Random Forest wins across all four metrics. An R^2 of 0.9934 means it explains over 99% of the NO_2 variation in the test set, and RMSE of 2.03 µg/m³ means predictions are typically off by just 2 micrograms, which is well within usable range for air quality decisions. 95.3% accuracy means almost 19 out of 20 predictions land within 10% of the actual reading.

XGBoost trains in 4.2 seconds vs Random Forest's 12.7 seconds, but since we retrain quarterly rather than in real time, this speed difference doesn't matter much in practice. Given the accuracy gap, Random Forest was chosen as the primary model for the dashboard. For O_3, both models perform lower: Random Forest $R^2 = 0.5869$, XGBoost

R^2 = 0.5560. This was expected. O_3 is a secondary pollutant that forms from photochemical reactions requiring ultraviolet data that our CPCB dataset doesn't include. This matches what the literature reports, so it's not a model failure but a data limitation.

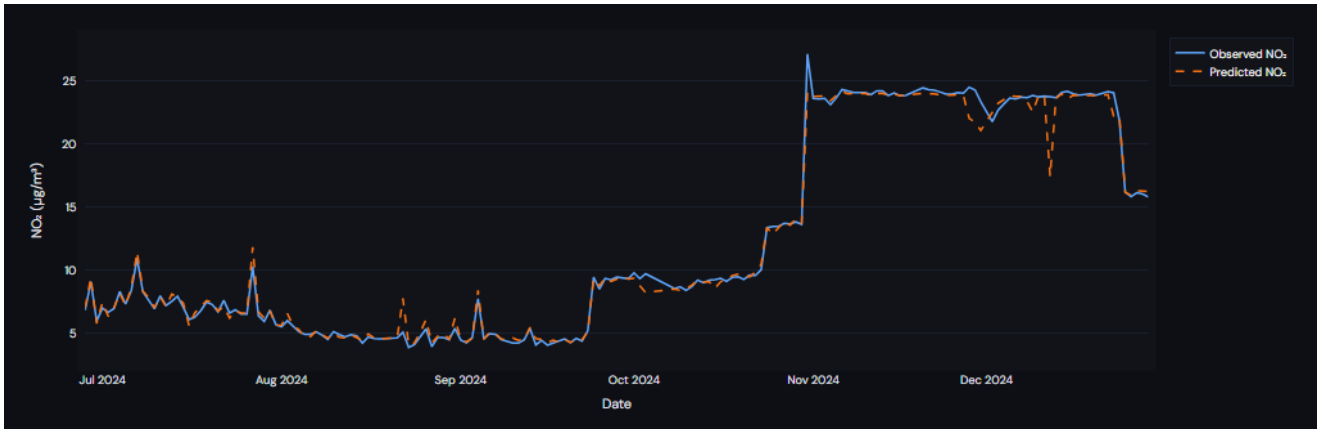

Fig 7.1. No_2 Observed Vs Predicted — Random Forest (2024)

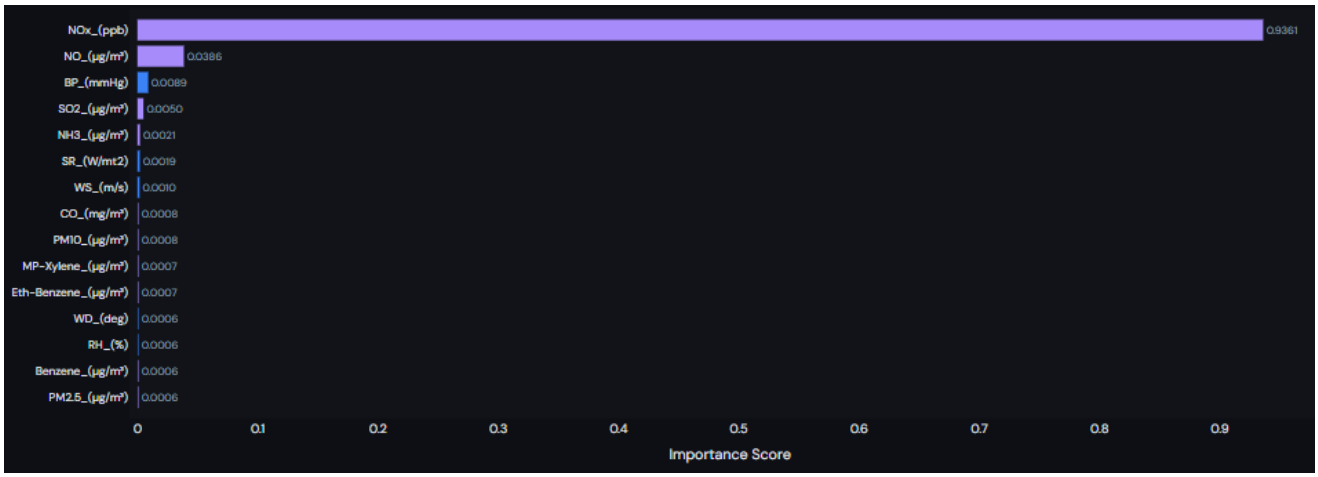

Fig 7.2. Feature Importance [Random Forest]

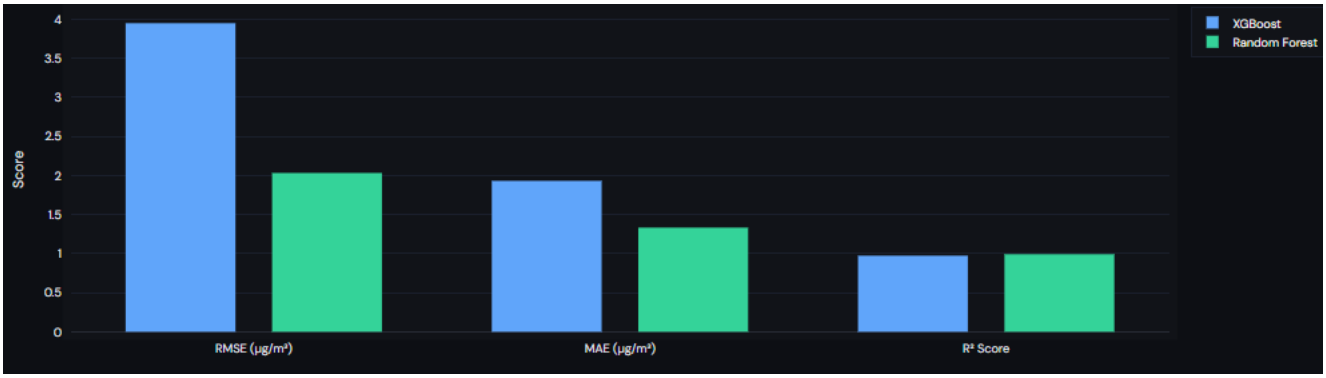

Fig 7.3. Model Comparison [Accuracy and Training Time]

References

[1] Kumar, K. and Pande, B.P. "Air pollution prediction with machine learning: a case study of Indian cities." International Journal of Environmental Science and Technology, vol. 20, pp. 5333–5348, 2023.

[2] Natarajan, S.K., Shanmurthy, P., Arockiam, D., Balusamy, B. and Selvarajan, S. "Optimized machine learning model for air quality index prediction in major cities in India." Scientific Reports, vol. 14, 2024.

[3] Boeing, G. "Modeling and Analyzing Urban Networks and Amenities with OSMnx." Geographical Analysis, 2025.

[4] Petrić, V., Hussain, H., Časni, K., Vuckovic, M. and Lovrić, M. "Ensemble Machine Learning, Deep Learning, and Time Series Forecasting: Improving Prediction Accuracy for Hourly Concentrations of Ambient Air Pollutants." Aerosol and Air Quality Research, vol. 24, 2024.

[5] Cabaneros, S.M.L.S., Calautit, J.K.S. and Hughes, B.R. "Hybrid Artificial Neural Network Models for Effective Prediction and Mitigation of Urban Roadside NO2 Pollution." Energy Procedia, vol. 142, pp. 3524–3530, 2017.

[6] Kothandaraman, D., Praveena, N., Varadarajkumar, K., Madhav Rao, B. and Dhabliya, D. "Intelligent Forecasting of Air Quality and Pollution Prediction Using Machine Learning." Computational Intelligence and Neuroscience, 2022.

[7] Gupta, A., Naidu, V.R. and Bhardwaj, R. "Air Quality Forecasting Using Machine Learning: Comparative Analysis and Ensemble Strategies for Enhanced Prediction." Water, Air, and Soil Pollution, 2025.

[8] Shakya, D., Deshpande, V., Goyal, M.K. and Agarwal, M. "PM2.5 air pollution prediction through deep learning using meteorological, vehicular, and emission data: A case study of New Delhi." Journal of Cleaner Production, vol. 427, 2023.

[9] Mandal, S., Boppani, S., Dasari, V. and Thakur, M. "A bivariate simultaneous pollutant forecasting approach by Unified Spectro-Spatial Graph Neural Network (USSGNN) for prediction of O3 and NO2 for New Delhi, India." Sustainable Cities and Society, vol. 114, 2024.

[10] Malhotra, M. and Aulakh, I.K. "Meteorological Factors Correlation with Air Pollutants: A Case Study in Delhi." International Journal of Environmental Science and Development, vol. 14, pp. 91–105, 2023.

[11] Rautela, K.S., Singh, S. and Goyal, M.K. "Characterizing the spatio-temporal distribution, detection, and prediction of aerosol atmospheric rivers on a global scale." Journal of Environmental Management, vol. 351, 2024.

[12] Krishan, M., Jha, S., Das, J., Singh, A., Goyal, M.K. and Sekar, C. "Air quality modelling using long short-term memory (LSTM) over NCT-Delhi, India." Air Quality, Atmosphere and Health, vol. 12, pp. 899–908, 2019.

[13] Mampitiya, L., Rathnayake, N., Leon, L.P., Mandala, V. and Rathnayake, U. "Machine Learning Techniques to Predict the Air Quality Using Meteorological Data in Two Stations in Sri Lanka." Environments, vol. 10, no. 8, p. 141, 2023.

[14] Breiman, L. "Random Forests." Machine Learning, vol. 45, pp. 5–32, 2001.

[15] Chen, T. and Guestrin, C. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.