

Multi-Stage Data Fusion Architecture for Urban Environmental Analytics: Integrating Predictive Regression and Hazard Anomaly Classification

Jayshree Ghorpade-Aher¹, Soham Kulkarni², Bala Jee³, Krishanu Roy⁴

^{1,2,3,4}DCET, SOCSE, Dr. Vishwanath Karad MIT WPU, Pune, India

Abstract—Air pollution is still one of the most serious challenges facing urban health and sustainability, but hourly readings from monitoring stations are seldom combined with the local circumstances that influence them. In this study, a multi-stage data fusion approach is presented which fuses Mumbai's hourly AQI and time-series data on pollutant levels PM_{2.5}, NO₂, SO₂, CO, and O₃ with a carefully prepared festival-anomaly table in order to carry out two separate prediction tasks: (i) the use of regression to estimate continuous PM_{2.5} concentrations, and (ii) classification to categorise hazard risk into five AQI bands (Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, and Very Unhealthy), with SMOTE employed to deal with the problem of class imbalance. Temporal features such as the PM_{2.5} value from the previous hour and moving average temperature are created, and all the models are tested using an 80/20 chronological split to avoid temporal leakage. For the regression task, Linear Regression, Random Forest, XGBoost, and LightGBM regressors are examined, while for the classification task, Random Forest, XGBoost, and LightGBM classifiers are assessed. The feature importance analysis indicates that the PM_{2.5} level from the previous hour, O₃, and CO are the main factors influencing pollution risk. The pipeline proposed here provides a simple, inexpensive, station-aware decision-support tool that increases the effective resolution of municipal air quality monitoring without the need for additional sensor equipment.

Index Terms—Air quality index (AQI), class imbalance, data fusion, feature importance, hazard classification, LightGBM, Mumbai, PM_{2.5} prediction, SMOTE, XGBoost.

I. INTRODUCTION

The concentrations of pollutants and the corresponding Air Quality Index (AQI) values produced by urban air quality monitoring networks are

given for each individual monitoring station, but the readings of a station are just as influenced by its local surroundings such as its proximity to traffic corridors, the type of land use area it is in, and the activities surrounding it as they are by the city-wide weather conditions. If the data from each station are considered without taking into account this topological context, then information that is directly relevant to both the level of pollution and its sudden rise into hazardous ranges is lost.

A. Motivation

Air pollution is one of the main urban issues concerning both sustainability and public health, since fine particulate matter (PM_{2.5}) is associated with respiratory and cardiovascular diseases at concentrations that are typical in the cities of India. Even though the monitoring stations in Mumbai located in commercial, residential and heavily trafficked areas are exposed to the same regional weather conditions they frequently record different pollution levels, indicating that the specific context of each station is a significant predictive factor that is ignored by general city-wide models.

From the point of view of sustainability, it is expensive to install and maintain densely deployed physical sensor networks, which in turn means that cities cannot monitor local pollution hotspots with great detail. A model that combines data from existing monitoring stations with low-cost, static topological metadata is able to improve the resolution of monitoring without having to make a proportional increase in infrastructure investment, thus directly contributing to both UN Sustainable Development Goal 11 (Sustainable Cities and Communities) and Goal 3 (Good Health and Well-Being). The ability to

predict the exact PM_{2.5} value enables detailed trend monitoring for long-term urban planning, while the ability to predict discrete hazard categories allows for clear, actionable alerts such as “Very Unhealthy” versus “Good” which both the public and local authorities can act on quickly in order to reduce exposure. Since severe pollution days are relatively rare in the historical record, correctly identifying them also constitutes a class imbalance problem in itself, and has real implications for public health and environmental equity if a severe day is overlooked.

In 2026 the situation has not improved in terms of the challenges involved. Even though carbon dioxide (CO₂) is not classified as a criteria air pollutant, it is the best indicator of the overall pattern of emissions: the UK Met Office predicts that the annual average concentration of CO₂ at Mauna Loa in 2026 will be 429.4±0.6 ppm, representing an increase of approximately 2.4 ppm on the 2025 figure, with the seasonal peak reaching near 432 ppm in May [10]. This rate of increase is still well in excess of what is consistent with limiting global warming to 1.5 °C, and the expected slight slowdown in 2026 is due solely to a temporary enhancement of natural carbon sinks caused by La Niña [10]. Since the sources that emit CO₂ also emit PM_{2.5}, carbon monoxide and precursors of ozone, and since a warmer climate is conducive to ozone formation and to stagnant air conditions, management of climate and air quality is becoming more and more linked. For rapidly growing cities in India the problem is made worse by the scarcity of regulatory monitoring stations and by occasional spikes, such as those that occur around major festivals, which require timely and locally appropriate warnings.

This is presented as an urban environmental analysis tool which is tailored to the local area: at its technical heart lies the combination of time series telemetry with static station metadata, the aim being to make the scale and the severity of pollution risk apparent earlier and more accurately than is possible through aggregate reporting at the city level, in order to help achieve more sustainable and equitable urban air quality management.

B. Problem Statement

Given hourly pollutant and weather reports for Mumbai, along with their static station-level

contextual information, the task is to (i) predict the hourly PM_{2.5} value for the given hour and (ii) classify the same hour's AQI into one of Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, and Very Unhealthy with an emphasis on the low-representation class Good. The two tasks are fed the same set of features to enable joint analysis of pollution levels and their severity. Note that this system is meant to be an aid to municipal monitoring and not a replacement for actual sensor data collection and regulatory reporting.

II. LITERATURE REVIEW

A substantial body of recent work applies machine learning to AQI estimation and classification, though most of it treats prediction as a single task conducted on pollutant and meteorological time series alone. Iqbal et al. [1] compare Linear Regression, Random Forest, XGBoost, and LSTM for AQI prediction across Delhi, Bengaluru, and Kolkata using PM_{2.5}/PM₁₀ and time features and evaluated on MAE, RMSE, and R², finding that tree-based ensembles consistently outperform the linear baseline. Karthick et al. [2] evaluate nine ML algorithms for AQI forecasting in Delhi, identifying Random Forest and XGBoost as top performers and PM_{2.5}/CO as the most influential features through feature importance analysis. Mandvi et al. [3] also compare XGBoost against Lasso regression for AQI prediction from PM₁₀, PM_{2.5}, SO₂, and NO₂, reporting substantially higher R² for XGBoost over the linear baseline, a pattern this project's own choice of XGBoost as the regressor is directly motivated by.

On the classification side, Setiawan et al. [4] employ a Random Forest classifier to measure and predict the Air Pollution Standard Index from O₃, CO, NO₂, SO₂, PM_{2.5}, and PM₁₀ concentrations at an airport monitoring station, finding the algorithm effective and that most days fall in the “Moderate” category, underscoring the class imbalance challenge this project's use of the SMOTE stage is motivated towards. Kristiyanto and Lubis [5] develop a SMOTE-based Random Forest classifier for a three-class (Good/Moderate/Unhealthy) air-quality index in South Tangerang, combined with SHAP-based explainability, reporting that hyperparameter-tuned, SMOTE-balanced Random Forest outperforms SVM and KNN baselines while surfacing SO₂ as the dominant driver of classification decisions; direct

precedent for pairing SMOTE-balanced Random Forest with feature-importance interpretation is what is being done here.

Data fusion approaches that go beyond time series of pollutant are comparatively rare, but promising. Porcheddu et al. [6] introduce a deep learning data-fusion method that combines low-resolution geophysical model data, high-resolution geographical indicators, ground station measurements, and satellite retrievals for estimating hourly PM_{2.5} maps at 100 m resolution over Paris, outperforming reanalysis estimates at hourly, daily, and monthly scales; direct precedent for fusing heterogeneous data sources rather than relying on pollutant time series is what is being done here. Ma et al. [7] combine hourly PM_{2.5} forecasts from an operational numerical weather model with near-surface pollutant and meteorological measurements to build an XGBoost model which produced daily PM_{2.5} forecasts with 50–100% higher correlation coefficients and 14–24 $\mu\text{g}/\text{m}^3$ lower standard deviation than the numerical model alone, illustrating the benefit of fusing complementary data sources before using them as inputs to a boosting model.

Exploring multi-source data fusion across heterogeneous topologies, Bansal et al. [8] evaluated machine learning and deep learning architectures including Random Forest, XGBoost, LSTM, and Transformers for AQI prediction in urban and industrial environments. By fusing air pollutant telemetry, meteorological indicators, satellite-derived aerosol optical depth, and industrial IoT sensor feeds they demonstrated that domain-specific dynamics govern model suitability. Transformer architectures yielded the highest predictive accuracy in urban zones due to their ability to model sequential temporal cycles and traffic rush hours. Conversely, TGNN models achieved higher accuracy in industrial zones by capturing spatial connectivity, inter-station dependencies, and point-source pollutant dispersion. Their SHAP interpretability findings established that while temporal features dominate urban AQI predictions, spatial proximity to emission sources are the dominant drivers in industrial zones. Directly corroborating the predictive value of topological and landmark-proximity metadata, the findings demonstrate why these factors could be beneficial for AQI prediction.

III. RESEARCH GAP

- Existing Indian-city AQI studies predominantly work at aggregate city/regional levels using pollutant and meteorological time series only, while ignoring local topological context such as proximity to major arterial highways, industrial zones, or coastal boundaries [1]–[3].
- While recently developed spatio-temporal frameworks such as Temporal Graph Neural Networks have modeled spatial correlations across monitoring grids [8], they require high density monitoring networks and substantial computational overhead. There exists a significant opportunity for lightweight station-aware data fusion pipelines that can include low-cost, static landmark-proximity metadata directly into standard tabular/tree-based learning frameworks.
- Most published work treats AQI prediction as either a regression task or a classification task, rather than jointly modelling both the continuous magnitude and the discrete hazard level from the same fused feature set.
- The class imbalance inherent to hazard categorization in this dataset (Good readings account for only 1.1% of records against a dominant 53% Moderate class) is frequently acknowledged but not consistently addressed with resampling techniques such as SMOTE in the Indian city AQI literature.

This project addresses that gap by building an end-to-end pipeline that fuses Mumbai's hourly AQI, then applies parallel regression and SMOTE-balanced classification stages on the same fused feature set.

IV. OBJECTIVES

1. The goal is to combine Mumbai's hourly AQI and pollutants time-series data into a fused data set.
2. Create time-based and contextual features such as Lag_PM_{2.5}, Rolling_Temp, Zone_Type, Season, and Distance_to_Highway.
3. Predict PM_{2.5} concentrations using Linear Regression, Random Forest, XGBoost, and LightGBM regressors, and predict AQI hazard classes using Random Forest, XGBoost, and LightGBM classifiers.

4. Address the class imbalance issue of AQI hazard classes using SMOTE.
5. Evaluation and interpretation of results using appropriate performance metrics and feature importance to understand the drivers of AQI and hazard risk.

V. DATASET

The project uses the “Air Quality Dataset: Indian Cities (2022–2025)” [19], a dataset on Kaggle, filtered by rows with City = Mumbai, resulting in 28,806 hourly observations between August 2022 and November 2025. This dataset has been merged with a manually created table, which contains distances from Mumbai’s coordinates to 5 selected landmarks to form a fused dataset used in all subsequent stages of modelling. Table I provides an overview of this dataset. Each row of the data is one hourly reading for Mumbai, with its date and time, the pollutant concentrations (NO₂, SO₂, CO, O₃), PM2.5, aerosol optical depth (AOD), and the air temperature at 2 m; the AQI category is derived from PM2.5. The landmark table is small and static, one row per landmark with its distance from the Mumbai coordinates. A third small table, a festival calendar with date and festival columns, was created by hand.

TABLE I Dataset Summary

Metric	Value
Base dataset	Air Quality Dataset: Indian Cities (2022–2025)
Filtered scope	Mumbai only
Train / test split	23,044 / 5,762 rows, chronological
Target (regression)	PM2_5_ugm3 concentration, µg/m ³ (continuous)
Target (classification)	AQI_Category, 5 classes: Good, Moderate, Unhealthy_Sensitive, Unhealthy, Very_Unhealthy
Class balance	Highly imbalanced (Moderate 53%, Good 1.1%); corrected via SMOTE on training folds only
Records used	28,806 hourly rows

A. Input Variables

The models use the pollutant, satellite, meteorological, temporal, and engineered data listed in Table II. The gaseous pollutants NO₂, SO₂, CO, and O₃ enter as co-

pollutant predictors of PM2.5 and of the hazard category; concentrations are in µg/m³.

TABLE II Input Variables Used in the Fused Dataset

Input group	Variables
Gaseous pollutants	Nitrogen dioxide (NO ₂), sulphur dioxide (SO ₂), carbon monoxide (CO), ozone (O ₃)
Particulate matter	PM2_5_ugm3 (regression target; its lagged and rolling values are used as inputs)
Satellite-derived	Aerosol optical depth (AOD)
Meteorological	Air temperature at 2 m (°C)
Temporal calendar	Hour, Day_of_Week, Season, festival-anomaly indicators (e.g., Diwali, Ganesh Chaturthi)
Engineered	Lag_PM2.5, 24-h rolling PM2.5 mean, Rolling_Temp (7-day mean)

VI. PROJECT ARCHITECTURE

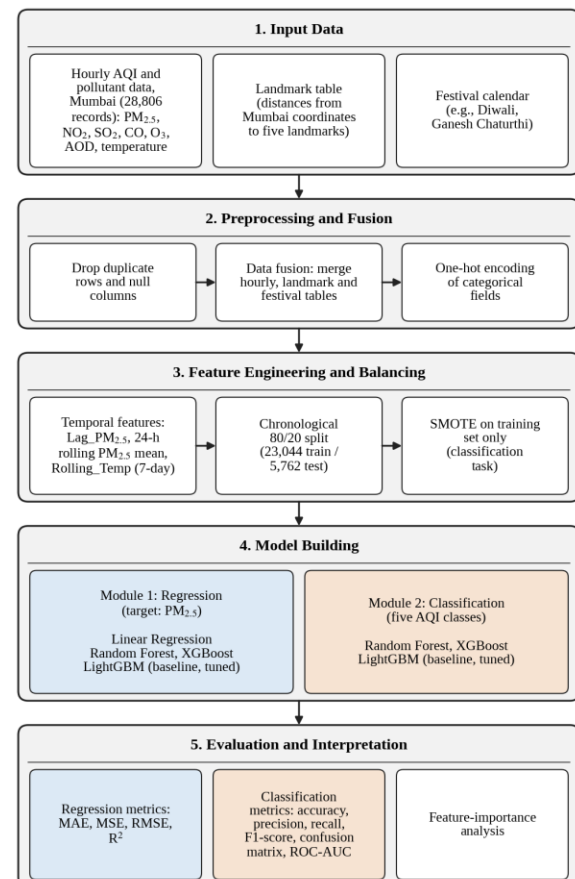


Fig. 1. Architecture of the proposed multi-stage data fusion pipeline.

Data flow top to bottom: input sources are cleaned and fused, engineered and balanced, fed to parallel regression and classification models, and evaluated in the proposed methodology.

This project designs an end-to-end machine learning pipeline for localized air-quality analytics in Mumbai. Hourly AQI and pollutant telemetry are fused with data on station context and transformed via temporal feature engineering and class balancing to yield two parallel modelling modules: regression and classification. The former estimates PM2.5 concentration while the latter assigns AQI hazard categories. Fig. 1 abstracts this sequence from input data through preprocessing, techniques, and model building to the proposed methodology; individual techniques are described in Section VII and the step-by-step methodology explained in Section VIII.

Data description. The pipeline works on three sources: (i) 28,806 hourly Mumbai records with pollutant concentrations, AOD, and 2 m air temperature; (ii) a static table of distances from Mumbai's coordinates to five landmarks; and (iii) a hand-built festival calendar. These are fused into one feature set that feeds both the regression task, whose target is the continuous PM2.5 concentration, and the classification task, whose target is the five-class AQI hazard category derived from PM2.5.

VII. TECHNIQUES AND ALGORITHMS

A. Feature Engineering

- One-Hot Encoding of categorical fields (Zone_Type, Day_of_Week, Season) to make station and temporal context usable by tree-based and linear models.
- Temporal feature engineering: Lag_PM2.5 (previous day's PM2.5) and Rolling_Temp (7-day rolling mean temperature) to capture short-term pollution persistence and seasonal drift.
- SMOTE (Synthetic Minority Over-sampling Technique) applied to the hazard class to correct class imbalance ahead of classifier training.

B. Module 1 – Regression

- Linear Regression as a baseline.
- Random Forest Regressor, XGBoost Regressor, and LightGBM Regressor (baseline and

hyperparameter-tuned) evaluated as ensemble alternatives.

C. Module 2 – Classification

- Random Forest, XGBoost, and LightGBM (baseline and hyperparameter-tuned) classifiers trained on the SMOTE-balanced fused dataset to predict the five-class AQI hazard label.

D. Algorithm Overview

This part introduces each algorithm in a few lines, followed by the scores it produced on the held-out test set. The same numbers appear in Tables III and IV and are discussed in Section X. All classifiers were trained on the SMOTE-balanced training data.

Linear Regression. This is the simplest model in the study and works as the baseline. It writes PM2.5 as a weighted sum of all the input features and picks the weights by ordinary least squares, meaning the weights that make the total squared error on the training data as small as possible. It is quick and easy to read, but it can only describe straight-line relationships. It was used for regression only and gave MAE 2.859, RMSE 3.835 and R^2 0.9509, the weakest result among the regressors.

Random Forest. A Random Forest grows many decision trees, each one on a random bootstrap sample of the training rows and with a random subset of features considered at every split. The trees are then combined by averaging their outputs for regression and by majority vote for classification. Because the trees differ from each other, their mistakes partly cancel out, which lowers the variance and makes overfitting less likely. As a regressor it reached MAE 1.507, RMSE 2.493 and R^2 0.9792. As a classifier it gave accuracy 0.9275, weighted precision 0.9271, weighted recall 0.9275 and weighted F1 0.9215, with a macro-AUC of 0.992. It was the weakest classifier mostly because of the good class, where it found only 40 of the 136 test hours (29.4% recall).

XGBoost. XGBoost (Extreme Gradient Boosting) also builds on decision trees, but it adds them one after another, and every new tree is fitted to correct the errors that the earlier trees left behind. A penalty on tree complexity and a learning rate that shrinks each step keep the model from overfitting. It was used here

with default settings. As a regressor it scored MAE 1.958, RMSE 3.141 and R^2 0.9670. As a classifier it was the best of the four, with accuracy 0.9509, weighted precision 0.9534, weighted recall 0.9509, weighted F1 0.9509 and macro-AUC 0.995, and it recovered 131 of the 136 Good hours (96.3% recall).

LightGBM (baseline). LightGBM is another gradient boosting library. It grows each tree leaf-wise, always splitting the leaf that lowers the loss the most, and it groups feature values into histogram bins, which makes training fast and light on memory. The baseline version uses the default hyperparameters. As a regressor it gave the lowest error of all models, MAE 1.506, RMSE 2.453 and R^2 0.9799. As a classifier it gave accuracy 0.9471, weighted precision 0.9486, weighted recall 0.9471, weighted F1 0.9473 and macro-AUC 0.994, with 129 of the 136 Good hours found (94.9% recall).

LightGBM (tuned). This is the same algorithm with its hyperparameters (number of estimators, number of leaves, maximum depth, learning rate and the subsample and column-sample ratios) searched by RandomizedSearchCV. The search used time-series cross-validation on the training part only, so no future data leaked into it. As a regressor it gave MAE 1.535, RMSE 2.457 and R^2 0.9798. As a classifier it gave accuracy 0.9467, weighted precision 0.9479, weighted recall 0.9467, weighted F1 0.9466 and macro-AUC 0.994, with 131 of the 136 Good hours found (96.3% recall). Tuning did not beat the baseline on either task, a point taken up in Section X.

VIII. METHODOLOGY

A. Data Acquisition

Download the Air Quality Dataset for Indian Cities (2022–2025) and filter to rows where City = Mumbai, then preprocess. The filtered file keeps the hourly timestamp, the pollutant, AOD and temperature columns, and the Mumbai-only rows form the base time series for the later steps.

B. Dataset Creation

Manually build a dataset to use for fusion for major festivals that create an anomaly spike, such as Diwali or Ganesh Chaturthi. This was done as a small pandas dataframe with two columns: date, holding the

calendar day of the festival, and festival, holding its name. Festivals were chosen because fireworks, large crowds, idol immersion traffic and heavy vehicle movement can push PM2.5 well above its usual level for a day or two, and the pollutant and weather columns alone do not explain such spikes. Matching the date column against the date of each hourly record lets every hour of a festival day carry a festival label, and the one-hot encoded labels become the festival-anomaly indicators listed in Table II. Hours that fall on no festival simply keep the value zero in those indicator columns.

C. Data Fusion

Join the filtered time-series AQI/weather data with the station metadata table using pandas merge () on Station_Name, producing a single relational dataset combining high-frequency telemetry with static topological context.

D. Data Preprocessing

Drop duplicate rows and remove any null columns. Merge all final datasets into one cleaned fused dataset.

E. Temporal Feature Engineering

Derive lag and rolling features from the fused series, including Lag_PM2.5 (previous day's PM2.5) and Rolling_Temp (7-day rolling mean temperature). Perform feature encoding, such as one-hot encoding of nominal columns, to convert them into numerical columns for data analysis.

F. Class Balancing

Bucket AQI into hazard categories; apply SMOTE to the minority classes so the classification model is not biased toward common categories.

G. Train/Test Split

Split the fused, encoded dataset chronologically: the earliest ~80% of hourly records (Aug 2022–Mar 2025) for training, the most recent ~20% (Mar–Nov 2025) held out for testing rather than a random split, since the lag and rolling features depend on time order and a random split would leak future information into training.

H. Model Training

Train a Linear Regression model as a baseline, then tree-based regressors (Random Forest, XGBoost, and

LightGBM), to predict PM2.5 from fused features. In parallel, train Random Forest, XGBoost, and LightGBM classifiers on the SMOTE-balanced data to predict the discrete hazard category.

I. Evaluation and Interpretability

Compare regression models using RMSE, MAE, and R^2 ; compare the classifiers using precision, recall, and F1-score. Extract feature importance from the tree-based models to identify the leading drivers of PM2.5 and hazard classification.

IX. EXPERIMENTATION

A. Experimental Setup

All experiments are implemented in Python using pandas and NumPy to handle data, scikit-learn to perform Linear Regression, fit a Random Forest, and compute evaluation metrics, the XGBoost library to fit a gradient-boosted regressor, and imbalanced-learn to apply SMOTE. The fused dataset is divided chronologically into training and hold-out test partitions (approximately 80% / 20%), so that both the regression and classification stages are evaluated on unseen data.

B. Model Configuration

The Linear Regression baseline model and the untuned Random Forest, XGBoost, and LightGBM regressors are fit using default or standard hyperparameter configurations, respectively. LightGBM Regressor and LightGBM Classifier are each tuned using RandomizedSearchCV on a grid search over key hyperparameters (number of estimators, number of leaves, max depth, learning rate, subsample/colsample ratios) using time-series cross-validation on the training partition, with the configuration minimizing cross-validated MAE or maximizing cross-validated weighted F1-score (average='weighted') being selected for final evaluation.

C. Class-Imbalance Handling

SMOTE is applied only to the training partition following the learning set/test set partition, ensuring that synthetic minority class instances are not introduced into the holdout set. The Random Forest Classifier is trained once on the original learning set, and once on the learning set with SMOTE-applied

oversampling, allowing for a direct comparison of minority-class recall between the two approaches.

D. Evaluation Protocol

The regression performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the determination coefficient (R^2) on the test set. For classification performance, precision, recall, F1-score, and confusion matrix were used.

Before the scores are reported, the metrics are defined below. For the regression task, let y_i be the observed PM2.5 value of the i -th hourly test record, $\hat{y}_i$ the value predicted by the model, $\bar{y}$ the mean of the observed test values and n the number of test records. The Mean Absolute Error (MAE) is the average size of the prediction error in $\mu\text{g}/\text{m}^3$, whichever direction it goes (1). The Mean Squared Error (MSE) squares every error before averaging, so a few big misses hurt the score much more than many small ones (2). The Root Mean Squared Error (RMSE) is just the square root of MSE, which puts the error back into $\mu\text{g}/\text{m}^3$ (3). The coefficient of determination (R^2) tells how much of the variation in PM2.5 the model manages to explain, with 1 meaning a perfect fit (4). MAE, RMSE and R^2 are listed in Table III, and MSE is plotted in Fig. 2.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

$$RMSE = \sqrt{MSE} \quad (3)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

For the classification task every hazard class c (Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy and Very Unhealthy) is taken in turn as the positive class against the other four. From the confusion matrix this gives the true positives (TP_c), false positives (FP_c) and false negatives (FN_c) of that class, and N is the total number of test records. Accuracy is the share of all hours whose class was predicted correctly (5). Precision of a class says how many of the hours the model labelled as that class really belong to it (6), and recall says how many of the hours that really belong to the class were found (7).

Recall is the one that matters most for the rare hazardous classes, because it shows how many bad days were missed. The F1-score is the harmonic mean

of precision and recall, so it is high only when both are high (8).

$$Accuracy = \frac{\sum_c TP_c}{N} \quad (5)$$

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \quad (6)$$

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \quad (7)$$

$$F1_c = \frac{2 \cdot Precision_c \cdot Recall_c}{Precision_c + Recall_c} \quad (8)$$

The five classes are very unevenly spread (Moderate is about 53% of the records and Good only 1.1%), so the per-class scores cannot simply be averaged with equal importance, and one overall number is still needed for comparing the models. For this reason, precision, recall and F1-score were calculated with average='weighted' in scikit-learn. Each class score is multiplied by its support n_c , the number of true test records of that class, and the sum is divided by N (9). A class therefore counts in proportion to how often it occurs. One side effect is that the weighted recall works out to the same value as the accuracy, which is why those two columns are identical in Table IV.

$$Score_{weighted} = \sum_c \frac{n_c}{N} Score_c \quad (9)$$

The confusion matrix itself (Fig. 6) lists the actual class in the rows and the predicted class in the columns, so the diagonal holds the correct predictions. The ROC curve plots the true positive rate (which is the recall) against the false positive rate, $FPR = FP / (FP + TN)$, while the decision threshold on the predicted class probability is moved from 0 to 1. The area under this curve (AUC) is 1 for a perfect ranking and 0.5 for random guessing (10).

Since there are five classes, ROC curves are drawn one-vs-rest, one per class (Fig. 7), and the macro-AUC is the simple mean of the five class AUCs (11).

$$AUC_c = \int_0^1 TPR_c(FPR) d(FPR) \quad (10)$$

$$Macro\ AUC = \frac{1}{C} \sum_{c=1}^C AUC_c \quad (11)$$

X. RESULTS AND PERFORMANCE ANALYSIS

What we noticed was that gradient-boosted and bagged tree ensembles outperformed the linear baseline on both tasks. For regression, LightGBM (baseline) gave the lowest error (MAE 1.506, R^2 0.9799) and Linear Regression the highest (MAE 2.859). For classification, XGBoost was best (accuracy 0.9509, weighted F1 0.9509) and Random

Forest weakest (accuracy 0.9275), mainly because it missed most of the rare good hours. Hyperparameter tuning of LightGBM gave no gain on either task. The subsections below give the full tables and figures.

A. Regression Performance

Table III contains the results for all regression models in terms of MAE, RMSE, and R^2 on the held-out test set. As can be seen, LightGBM (baseline) achieved the lowest error, slightly preceding the tuned version and the Random Forest Regressor.

On the contrary, in the case of the XGBoost Regressor, the error is considerably higher than that of the other tree-based ensemble models, which contradicts the findings in the literature reviewed in Section II [1], [3]. The graphical comparison of the results is given in Fig. 2 and Fig. 3.

TABLE III Regression Performance Comparison (MAE and RMSE in $\mu g/m^3$)

Model	MAE	RMSE	R^2
Linear Regression	2.859	3.835	0.9509
Random Forest Regressor	1.507	2.493	0.9792
XGBoost Regressor	1.958	3.141	0.9670
LightGBM Regressor (Baseline)	1.506	2.453	0.9799
LightGBM Regressor (Tuned)	1.535	2.457	0.9798

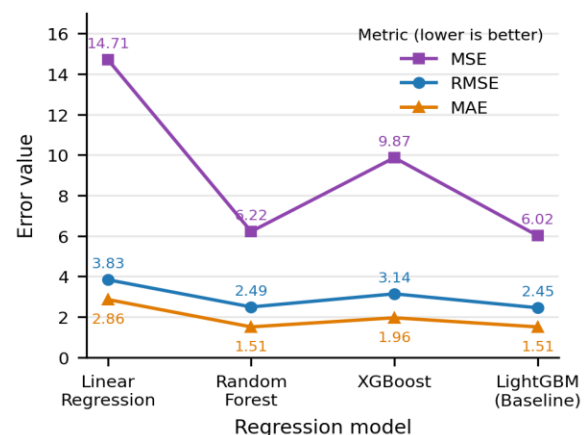


Fig. 2. MAE, MSE, and RMSE of the four base regression models (lower is better). Linear Regression has the highest error on every metric; Random Forest and LightGBM have the lowest.

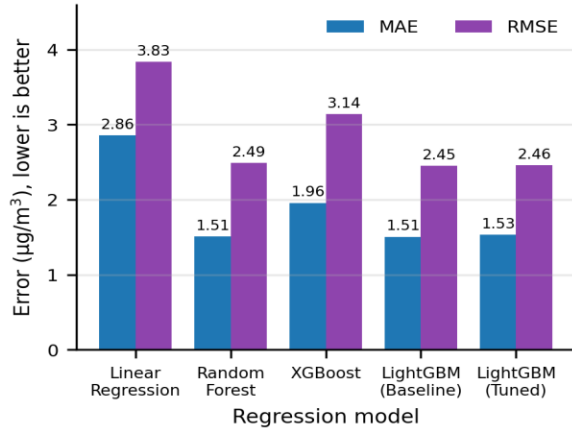


Fig. 3. MAE and RMSE for all five regression configurations ($\mu\text{g}/\text{m}^3$). Both LightGBM variants and Random Forest are almost identical, with XGBoost and Linear Regression clearly behind.

B. Classification Performance

Table IV reports accuracy and weighted precision, recall, and F1-score for the classification algorithms after SMOTE balancing. Recall on the rare, hazardous classes is the headline metric for this project, since an early-warning system that silently misses hazardous days provides limited practical value regardless of its overall accuracy.

Fig. 4 and Fig. 5 compare the classifiers. Because the classes are imbalanced, precision, recall and F1-score here are weighted averages (average='weighted', (9)), so every class counts in proportion to its number of test records.

TABLE IV Classification Performance (Weighted Averages, Post-SMOTE)

Model	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.9275	0.9271	0.9275	0.9215
XGBoost Classifier	0.9509	0.9534	0.9509	0.9509
LightGBM Classifier (Baseline)	0.9471	0.9486	0.9471	0.9473
LightGBM Classifier (Tuned)	0.9467	0.9479	0.9467	0.9466

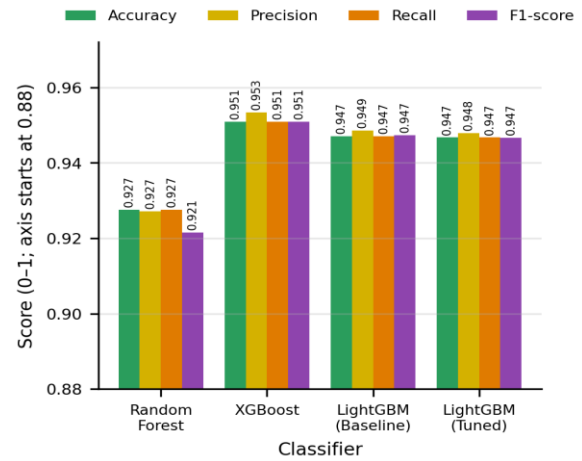


Fig. 4. Accuracy and weighted precision, recall, and F1-score of the four classifiers after SMOTE. XGBoost scores highest on every metric and Random Forest lowest, mainly on F1. The y-axis starts at 0.88 so that small differences are visible.

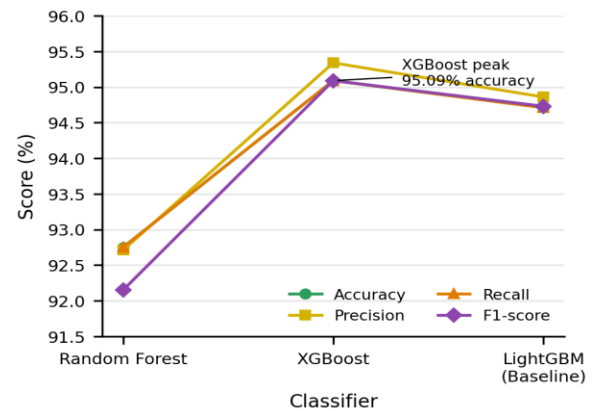


Fig. 5. The same classification metrics for the three model families (LightGBM shown as baseline). Scores rise from Random Forest to a peak at XGBoost and fall slightly for LightGBM.

C. Feature Importance

Feature-importance scores derived from the optimized LightGBM Regressor reveal that Lag_1hr_PM2.5 takes precedence by a considerable margin, followed in turn by the O₃, CO, AOD, and NO₂ concentrations, i.e., the recent pollution trend and co-pollutant levels dominate as the primary predictors with the proximity-based features displaying a comparatively lesser influence than initially postulated (Fig. 9 and Fig. 10).

The confusion matrices of the four classifiers are shown in Fig. 6. Their ROC curves are shown per class in Fig. 7 and overall, in Fig. 8.

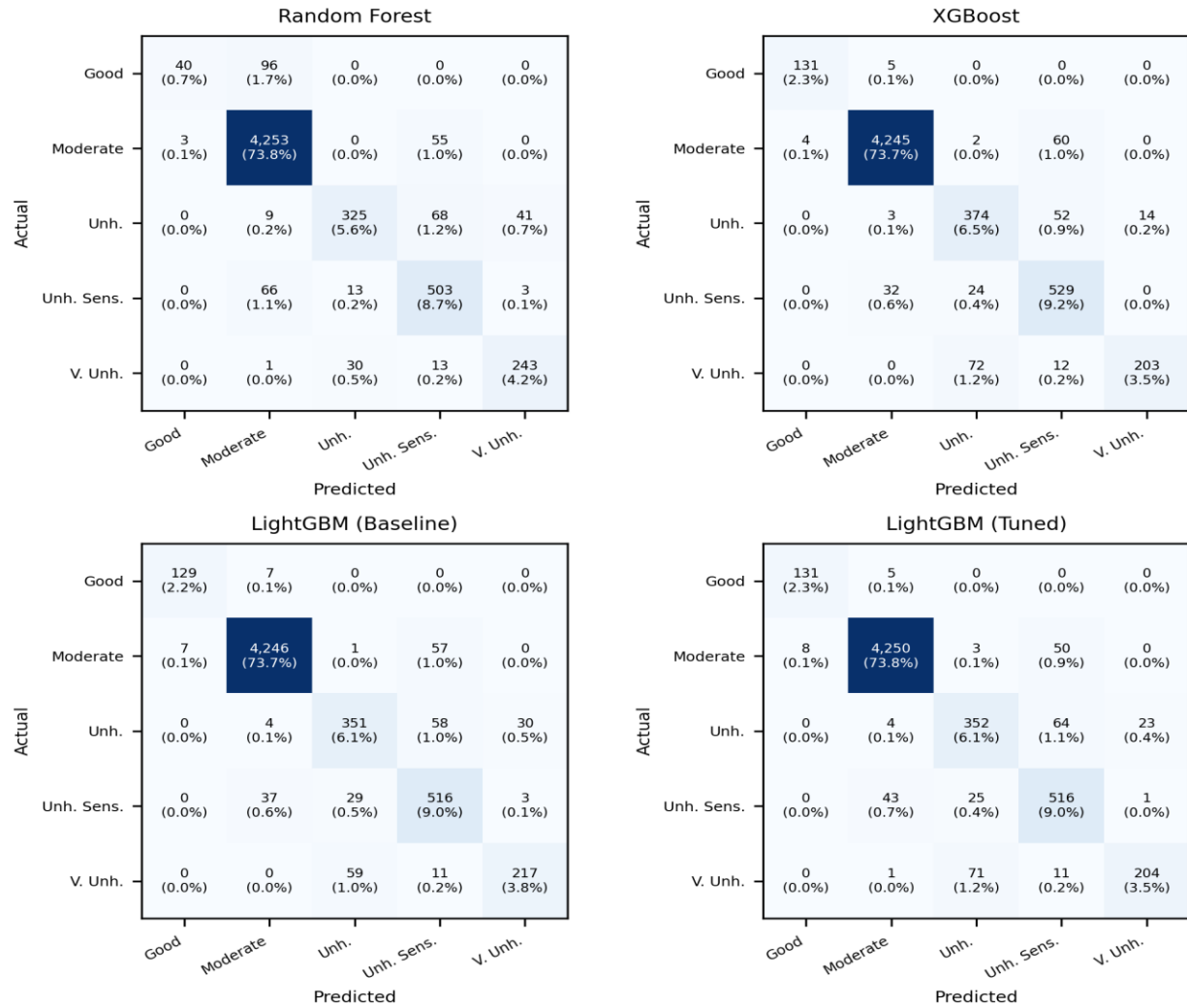
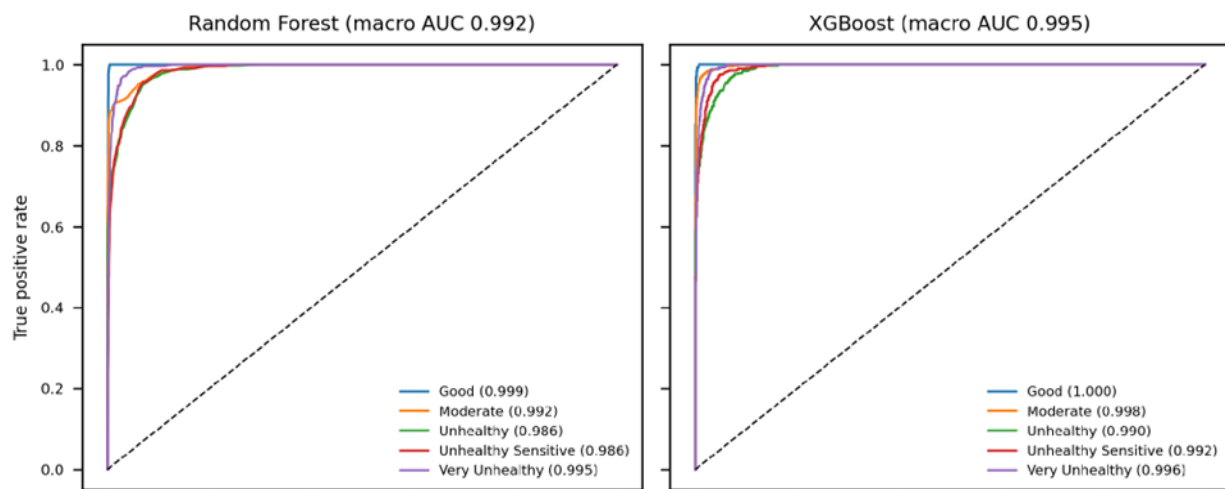


Fig. 6. Confusion matrices of the four classifiers on the hold-out test set (rows: actual class; columns: predicted class; each cell shows the count and its share of all 5,762 test hours). XGBoost and both LightGBM models recover the rare good class far better than Random Forest.



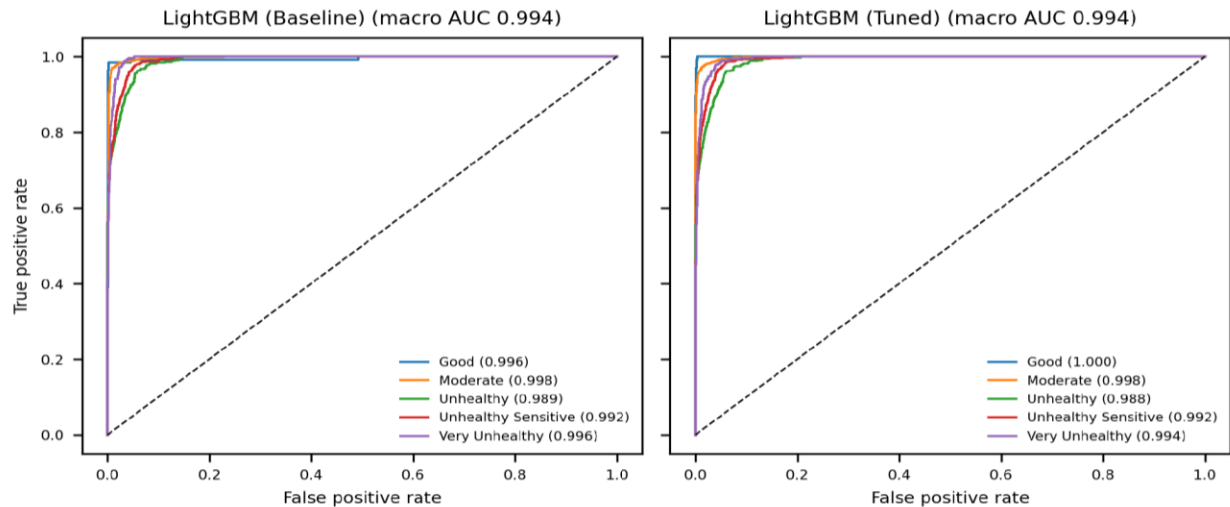


Fig. 7. Per-class (one-vs-rest) ROC curves for each classifier, with the AUC of each class in the legend and the macro-averaged AUC in each panel title.

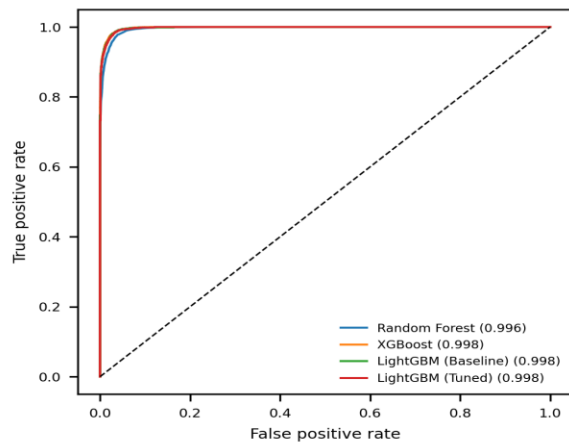


Fig. 8. ROC curves of the four classifiers on the hold-out set, with AUC in parentheses. The dashed diagonal marks chance-level performance.

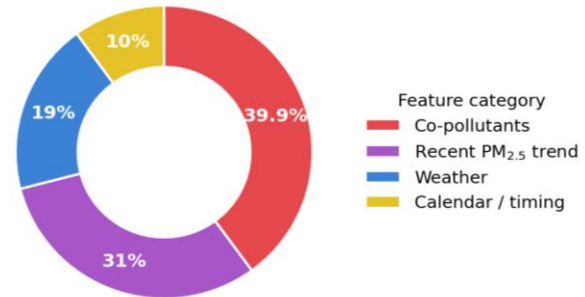


Fig. 10. Share of total importance by feature category: co-pollutants 39.9%, recent PM_{2.5} trend 31%, weather 19%, and calendar/timing 10%.

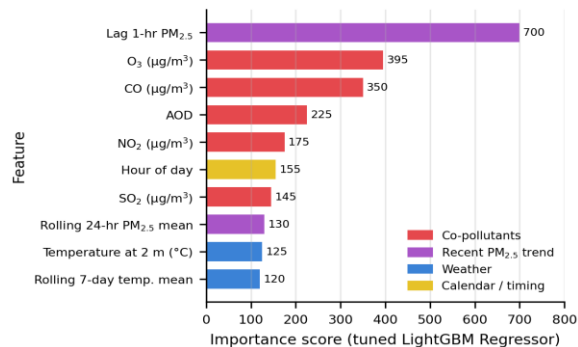


Fig. 9. The ten most important features of the tuned LightGBM Regressor, coloured by category. Previous-hour PM_{2.5} is far ahead, followed by O₃ and CO.

Why the top three features matter. (1) Lag PM_{2.5}: hourly PM_{2.5} is strongly autocorrelated, as fine particles accumulate under stagnant conditions and are slowly dispersed and deposited, so the previous reading summarizes best the current atmospheric state. (2) O₃: ozone is a secondary pollutant formed photochemically from nitrogen oxides and VOCs, while the same photochemistry leads to secondary aerosol formation, thus, O₃ carries information about the atmospheric oxidation and weather conditions that primary-emission indicators miss. (3) CO: carbon monoxide is a tracer of combustion sources, including vehicles, biomass, and waste burning, which are also major sources of PM_{2.5}, thus, CO is emitted along with PM_{2.5}. These are possible mechanistic explanations consistent with the results, but not causal findings of this study.

The relevance of these variables is reflected in the standards set by regulators. The WHO Global Air Quality Guidelines (2021) specify guideline concentrations for PM_{2.5}, PM₁₀, O₃, NO₂, SO₂, and CO, among others, including 5 µg/m³ (annual) and 15 µg/m³ (24-hour) for PM_{2.5} [11]. The U.S. Environmental Protection Agency sets National Ambient Air Quality Standards for the criteria pollutants, including PM_{2.5}, O₃, CO, NO₂, and SO₂, and in 2024 tightened the annual PM_{2.5} standard to 9.0 µg/m³ [12]. In India, the Central Pollution Control Board's National Ambient Air Quality Standards and National Air Quality Index likewise cover PM_{2.5}, O₃, CO, NO₂, and SO₂ [13].

D. Model Comparison

Overall, in both prediction tasks, gradient-boosted ensembles produced better results than the linear baseline, but for neither task did a single algorithm win decisively, as XGBoost was the most consistent choice in the previous literature. For the regression, LightGBM (baseline) yielded the lowest error (MAE 1.506, R² 0.9799), followed by Random Forest (MAE 1.507, R² 0.9792). Surprisingly, XGBoost Regressor performed considerably worse (MAE 1.958, R² 0.9670). RandomizedSearchCV optimization of LightGBM parameters produced no gains over the default baseline, suggesting the default hyperparameters were near optimal for the present feature set. On the other hand, the performance ranking flipped on the classification task: XGBoost had by far the best classifier accuracy (0.9509, 0.9509 weighted F1), followed by the two LightGBM alternatives. The accuracy of the XGBoost classifier was markedly better than Random Forest, which had a skewed distribution of errors, having performed worst on the minority good class.

This finding aligns with the recent observation of Bansal et al. [8], that while tree-based ensembles (Random Forest and XGBoost) offer reliable baselines for tabular data, specialized architectures are necessary to tease out temporal recurrence from spatial dispersion. Though deep sequential and graph networks attain high precision at the cost of substantial compute overhead [8], the multi-stage architecture presented in this work achieves comparable continuous estimation and discrete hazard categorization through the use of static spatial fusion and standard ensemble boosting.

E. Why Some Models Performed Better

Regression. Linear Regression trailed (MAE 2.859, R² 0.9509) because PM_{2.5} has non-linear dependencies and interactions (e.g., lag PM_{2.5}_ugm's impact is moderated by seasonality, time of day, and other pollutants) that a linear surface cannot capture, whereas trees can inherently model these relationships by branching on these variables. LightGBM and Random Forest regressed better, probably because the target variable has a strong signal (in the form of lag PM_{2.5}_ugm) that is easily learned by an ensemble of trees, especially in a relatively small table. Random Forest's bagging reduces variance, being an average of many independent deep trees, and LightGBM's histogram-based leaf-wise boosting also has advantages for structured data. We think XGBoost underperformed (MAE 1.958) probably not because of its algorithmic shortcomings but due to default hyperparameters that were not tuned. Since tuning had little benefit for LightGBM, it makes sense to try adjusting the hyperparameters for XGBoost, such as tree depth, learning rate, and regularization.

Classification. XGBoost was probably best because regularized sequentially corrected trees have better adaptation ability to the SMOTE-balanced data and mixed one-hot encoded and continuous features than Random Forest's bagging approach. Random Forest's problem was class Good with only 1.1% of cases: its trees vote for classes based on majority, but there is not much real data to learn how to classify such items, and interpolated SMOTE samples may not be representative enough, while boosting reweights those hard-to-classify instances at each step. Tuned LightGBM slightly underperformed its baseline (accuracy 0.9467 vs. 0.9471) because hyperparameters were selected by time-series cross validation on earlier data, and March–November 2025 has different seasonal trends. These are plausible explanations; ablation experiments would be needed to confirm these theories.

XI. CONCLUSION

This project outlines a multi-stage data fusion architecture which combines Mumbai hourly AQI telemetry for continuous PM_{2.5} regression and discrete hazard classification. By jointly fusing time-series pollutant data with static station metadata and

applying SMOTE to correct for class imbalance in the hazard classes, the pipeline aims to fill a gap in the current literature, which rarely combines joint regression/classification models. The resulting system is designed to be an inexpensive, station-aware decision tool, which extends the practical resolution of municipal air-quality monitoring without additional physical sensor infrastructure, in direct support of sustainable and equitable urban environmental management. Future work includes extending the fused dataset beyond Mumbai to other Indian metropolitan regions to assess geographic generalizability.

REFERENCES

- [1] Iqbal, P. Sarkar, and N. Mukherjee, "Comparative analysis of machine learning models for AQI prediction in Indian metro cities," *International Journal of Engineering Research & Technology*, vol. 13, no. 12, 2024, doi: 10.5281/zenodo.18130423.
- [2] K. Karthick, S. K. Aruna, R. Dharmaprasanth, and G. Ravindran, "Integrating machine learning techniques for air quality index forecasting and insights from pollutant-meteorological dynamics in sustainable urban environments," *Earth Science Informatics*, vol. 17, no. 4, pp. 3733–3748, 2024, doi: 10.1007/s12145-024-01382-8.
- [3] Mandvi, P. K. Patel, and H. K. Singh, "Performance analysis of machine learning models for AQI prediction in Gorakhpur City: A critical study," *Environmental Monitoring and Assessment*, vol. 196, no. 10, 2024, doi: 10.1007/s10661-024-13107-x.
- [4] A. Setiawan, U. L. Wibowo, A. Mubarak, K. Larasati, and J. A. H. Hammad, "Random Forest algorithm to measure the air pollution standard index," *Knowledge Engineering and Data Science*, vol. 7, no. 1, pp. 86–100, 2024, doi: 10.17977/um018v7i12024p86-100.
- [5] A. Kristiyanto and H. Lubis, "Development of an air quality classification system using SMOTE-based Random Forest and XAI analysis," *Zetroem*, vol. 8, no. 1, pp. 26–35, 2026, doi: 10.36526/ztr.v8i1.7586.
- [6] A. Porcheddu, K. Ville, L. Timo, and A. Lipponen, "Machine learning data fusion for high spatio-temporal resolution PM2.5," *EGUsphere*, pp. 1–23, 2025, doi: 10.5194/egusphere-2024-4056.
- [7] J. Ma, Z. Yu, Y. Qu, J. Xu, and Y. Cao, "Application of the XGBoost machine learning method in PM2.5 prediction: A case study of Shanghai," *Aerosol and Air Quality Research*, vol. 20, no. 1, pp. 128–138, 2020, doi: 10.4209/aaqr.2019.08.0408.
- [8] S. K. Bansal, S. R. Avula, M. A. Mehrotra, P. Singh, A. K. Gupta, and P. Ghosh, "Machine learning algorithms for predicting air quality index: A case study in urban and industrial zones," *Periodicals of Engineering and Natural Sciences*, vol. 13, no. 2, pp. 425–434, Jun. 2025, doi: 10.21533/pen.v13i2.354.
- [9] R. Muralikrishnan, S. Gollapalli, E. Sellappan, J. Prasanna, V. Murali, and A. Vijayakumar, "Machine learning-driven prediction and interpretation of air quality index in industrial environment," *Asian Journal of Civil Engineering*, vol. 27, no. 3, pp. 1473–1491, Nov. 2025, doi: 10.1007/s42107-025-01572-9.
- [10] Met Office, "Mauna Loa carbon dioxide forecast for 2026," 2026. [Online]. Available: <https://www.metoffice.gov.uk/research/climate/seasonal-to-decadal/seasonal-forecast/forecasts/co2-forecast>.
- [11] World Health Organization, WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide. Geneva, Switzerland: WHO, 2021.
- [12] U.S. Environmental Protection Agency, "Reconsideration of the National Ambient Air Quality Standards for Particulate Matter," *Federal Register*, vol. 89, Mar. 2024.
- [13] Central Pollution Control Board, National Air Quality Index. New Delhi, India: Ministry of Environment, Forest and Climate Change, 2014.
- [14] J. Ghorpade-Aher, A. Saxena, M. Inamdar, A. Thakre, D. Sinha, and T. Thesya, "QSAR model for aquatic toxicity estimates using machine learning," *Lecture Notes in Electrical Engineering*, pp. 387–398, 2025, doi: 10.1007/978-981-96-4679-1_31.
- [15] G. Ravindran, G. Hayder, K. Kanagarathinam, A. Alagumalai, and C. Sonne, "Air quality prediction by machine learning models: A predictive study on the Indian coastal city of Visakhapatnam,"

- Chemosphere, vol. 338, Art. no. 139518, 2023, doi: 10.1016/j.chemosphere.2023.139518.
- [16] F. Illescas-Martinez, L. Garcia, A.-J. Garcia-Sanchez, R. Asorey-Cacheda, and J. Garcia-Haro, "Air quality forecasting in non-monitored urban areas through machine and deep-learning model," *Expert Systems with Applications*, vol. 284, 2025, doi: 10.1016/j.eswa.2025.127749.
- [17] G. Ravindiran, S. Rajamanickam, K. Kanagarathinam, G. Hayder, G. Janardhan, P. Arunkumar, S. Arunachalam, A. A. AlObaid, I. Warad, and S. K. Muniasamy, "Impact of air pollutants on climate change and prediction of air quality index using machine learning models," *Environmental Research*, vol. 239, Art. no. 117354, 2023, doi: 10.1016/j.envres.2023.117354.
- [18] H. Tao, A. H. Jawad, A. H. Shather, Z. Al-Khafaji, T. A. Rashid, M. Ali, N. Al-Ansari, H. A. Marhoon, S. Shahid, and Z. M. Yaseen, "Machine learning algorithms for high-resolution prediction of spatiotemporal distribution of air pollution from meteorological and soil parameters," *Environment International*, Art. no. 107931, 2023, doi: 10.1016/j.envint.2023.107931.
- [19] B. Vekariya, "Air Quality Dataset: Indian Cities (2022–2025)," Kaggle, 2025. [Online]. Available: <https://www.kaggle.com/datasets/bhautikvekariya21/air-quality-dataset-indian-cities-2022-2025>.