IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

AI-Driven Air Quality Forecasting and Smart Decision Support Dashboard

Madhuri P. Borawake[1], Shruti Santosh Thorat[2], Pranita Pradip Lohokare[3], Nandini Nathuram Mahadik[4]

Head of Department and Associate Professor, Department of Computer Engineering, Pune District Education Association's College of Engineering, Pune, India[1]

Student, Department of Computer Engineering, Pune District Education Association's College of Engineering, Pune, India[2]

Student, Department of Computer Engineering, Pune District Education Association's College of Engineering, Pune, India[3]

Student, Department of Computer Engineering, Pune District Education Association's College of Engineering, Pune, India[4]

Abstract: Urban air pollution threatens public health, yet most monitoring services only report what the air is like right now. This paper presents the design of a web-based platform that goes one step further by estimating the Air Quality Index (AQI) for the coming 72 hours and turning those estimates into practical guidance. The platform merges pollutant readings obtained from OpenAQ and the Central Pollution Control Board (CPCB) with weather variables such as temperature, humidity, wind speed and rainfall. After cleaning and feature construction, an XGBoost regression model learns the link between these inputs and future AQI values. Training, tuning and testing follow a strictly time-ordered split so that no future information leaks into the learning stage. Forecasts are mapped to Indian AQI categories and paired with health advisories, rule-based recommendations and a question-answering chatbot, all delivered through an interactive dashboard built with FastAPI, React.js and PostgreSQL. The system is still being implemented, so no accuracy figures are claimed here; the paper instead documents the architecture, data plan and an evaluation protocol based on MAE, RMSE and R^2.

Keywords: Air Quality Index; AQI Forecasting; XGBoost Regression; OpenAQ; CPCB; Decision Support; Health Advisory; Chatbot; Environmental Monitoring

I. INTRODUCTION

Polluted air is now one of the most visible environmental problems in fast-growing cities. Traffic emissions, industrial output, construction dust and the burning of waste or biomass all raise the concentration of harmful particles and gases in the atmosphere. Long exposure is linked to breathing disorders, heart disease and other chronic conditions, which is why tracking and anticipating pollution has become a public-health concern as much as an environmental one. The Air Quality Index (AQI) condenses several pollutant concentrations into a single number and a colour-coded category that ordinary citizens can read. A present-day AQI value, however, only tells people what has already happened. Someone planning a morning run, a school arranging outdoor events, or a municipal office deciding on dust-control measures needs to know what the next few days are likely to bring.

Machine learning offers a way to learn recurring patterns from past observations and project them forward. Typical inputs for such forecasts include earlier pollutant readings, previous AQI values and weather conditions such as temperature, humidity, wind speed and rainfall. Forecast quality, nevertheless, hinges on how complete, consistent and accurate the underlying records are. Gradient-boosted decision trees, and XGBoost in particular, handle tabular data with non-linear relationships well and are therefore a sensible candidate for this task [1], [3].

This paper describes a proposed platform, the *AI-Driven Air Quality Forecasting and Smart Decision Support Dashboard*, that combines pollutant data from OpenAQ and CPCB with weather data to estimate AQI for up to 72 hours ahead. The estimates are processed by a rules layer that attaches AQI categories, health advice and recommendations, and are shown on a web dashboard together with a project-specific chatbot.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

The contribution is a single application that joins four things that are usually offered separately: short-term AQI forecasting, easy-to-read visualisation, health-oriented guidance and conversational access. The system is under development; this paper therefore reports the design, the data plan and the evaluation protocol, and deliberately leaves out numerical accuracy results until testing is complete.

II. RELATED WORK

Many studies have applied machine learning to estimate pollutant concentrations and AQI values. Ravindiran et al. compared several learning models for AQI prediction and examined how air pollutants relate to climate change [2]. Charjan et al. built an ensemble of tree-based learners around carefully engineered features and showed that the choice of input variables strongly affects forecast quality [1].

Chen and Guestrin introduced XGBoost as a scalable implementation of gradient tree boosting that trains efficiently on structured data [3]. Because it captures non-linear interactions without heavy pre-processing, it suits environmental series that mix pollutant, weather and calendar variables.

Deep sequence models have also been tried. Mohapatra et al. combined convolutional and recurrent LSTM layers for AQI forecasting and then used SHAP, LIME and partial-dependence plots to explain what drove the predictions [4]. Such models can represent intricate temporal behaviour, but they generally demand more data and computing effort, so their benefit depends on the application.

On the data side, OpenAQ exposes air-quality measurements from many monitoring networks through an open API [5], CPCB publishes the National Air Quality Index and the Indian AQI categories [6], and Open-Meteo provides historical and forecast weather data through a free interface [7]. Building on these resources, the present work concentrates on an XGBoost-based 72-hour forecast embedded in a complete decision-support application rather than on a model alone.

III. PROPOSED SYSTEM

The platform covers the full path from raw data to user-facing advice inside one web application. Observations from OpenAQ and CPCB are joined with weather records, cleaned, and converted into model inputs. An XGBoost regressor produces the forecasts, which are then classified into AQI categories and translated into health messages and recommendations. Citizens and authorities reach all of this through a dashboard and a chatbot.

A. System Architecture

Fig. 1 shows the layered architecture. The *data source layer* holds the Weather API, OpenAQ and CPCB. In the *data processing layer*, the data pass through collection and integration, cleaning and pre-processing, feature engineering and storage. The prepared data feed the XGBoost model in the *machine learning layer*. The *interpretation and decision-support layer* contains the chatbot and the AQI category mapping, which in turn drives the health advisories and decision-support recommendations. The *application layer* consists of the FastAPI backend, the chatbot interface and the React.js web dashboard, and a *database layer* based on PostgreSQL keeps historical observations, forecasts and system records.

Three user roles are served, all reaching the system through the application layer. Citizens look up current and predicted AQI, read health advice and ask the chatbot questions. Authorities follow pollution trends and review the recommendations when planning responses. Administrators maintain data sources and system settings.

B. Data Collection Module

This module retrieves available pollutant measurements from OpenAQ and CPCB and weather attributes (temperature, humidity, wind speed and precipitation) from a weather API. Records are keyed by time and monitoring location so that air-quality and weather values can later be matched. How much data is available, and how often it is refreshed, depends on the chosen source and station.

C. Pre-processing and Feature Engineering

Incoming records are screened for gaps, duplicates, timestamp mismatches and physically implausible readings. Units and time zones are harmonised, and missing values are treated with methods appropriate to their pattern. Feature engineering then derives model inputs, which may include lagged AQI and pollutant values, rolling averages, weather variables and calendar features such as hour of day and day of week. The final feature list will be fixed once the collected data has been inspected.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

D. XGBoost-Based Forecasting

XGBoost regression builds an additive ensemble of decision trees, where each new tree corrects the residual error of the previous ones. Here it is trained to map the engineered features to future AQI values. A forecasting strategy that reaches 72 hours will be selected from the usual options, such as recursive prediction or a separate target for each horizon, according to validation error and the data available.

The dataset is partitioned in time order into training, validation and test portions so that the model never sees information from the future during learning. Hyper-parameters are tuned on the validation portion only.

E. AQI Interpretation and Health Advisory

Every forecast value is classified into one of the six Indian AQI categories defined by CPCB [6]: Good, Satisfactory, Moderate, Poor, Very Poor and Severe. Each category is linked to a short advisory. For instance, when a poor-air period is predicted, the dashboard suggests reducing strenuous outdoor activity and following official health guidance.

F. Decision Support and Chatbot

The decision-support module applies predefined rules to predicted conditions. Citizens receive precautionary suggestions, while authorities can examine trends and weigh suitable pollution-control steps. The chatbot answers questions about current AQI, forecasts, pollutants, weather and health precautions using only the data held by the system and a set of predefined responses; a lightweight intent classifier may be used to recognise frequent question types. When the requested information is not available, the chatbot is designed to say so rather than invent an answer.

G. Dashboard, Backend and Database

The React.js dashboard presents current AQI, forecast curves, historical trends, weather information and advisories through charts and summary indicators. A FastAPI backend links the dashboard to the model and to PostgreSQL, which stores observations, processed data and forecast records for retrieval and display.

H. System Workflow

The operating cycle is: (i) collect air-quality and weather data; (ii) clean it and build features; (iii) generate the 72-hour AQI forecast with the trained model; (iv) assign categories and create advisories and recommendations; and (v) publish the output on the dashboard and make it available to the chatbot.

I. Proposed Algorithm

Algorithm 1 gives a simple outline of the procedure followed by the platform. Steps 1–5 prepare, tune and test the forecasting model, and steps 6–9 describe the recurring cycle that produces forecasts, advisories and chatbot answers.

Algorithm 1: 72-hour AQI forecasting and advisory generation

Input:

Air-quality data from OpenAQ and CPCB Weather data
Selected monitoring location L Forecast horizon H = 72 hours

Expected Output:

Hourly AQI forecasts
AQI categories and health advisories Dashboard information and chatbot responses

1. Initialize the system and connect to the required data sources and PostgreSQL database.
2. Collect data from OpenAQ, CPCB, and weather APIs, including pollutant concentrations, temperature, humidity, wind speed, and rainfall.
3. Preprocess the data by removing duplicates, handling missing values, standardizing timestamps, and aligning records by location and time.
4. Perform feature engineering by generating lag features, rolling averages, weather variables, and time-based features.
5. Split the dataset chronologically into training, validation, and testing sets.
6. Train the XGBoost regression model using training data and tune its hyperparameters using validation data.
7. Evaluate the model on test data using MAE, RMSE, and R^2 metrics.
8. Generate AQI forecasts for each hour of the next 72 hours using the trained model and latest available inputs.
9. Classify predicted AQI values into Good, Satisfactory, Moderate, Poor, Very Poor, and Severe categories

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

according to CPCB rules.

10. Generate health advisories and recommendations based on the predicted AQI categories.
11. Store the results in PostgreSQL through the FastAPI backend.
12. Display the results on the React.js dashboard using forecast charts, pollution trends, and health advisories.
13. Process chatbot queries using stored forecasts and predefined response rules.
14. Repeat the forecasting process whenever new data become available and update the dashboard.

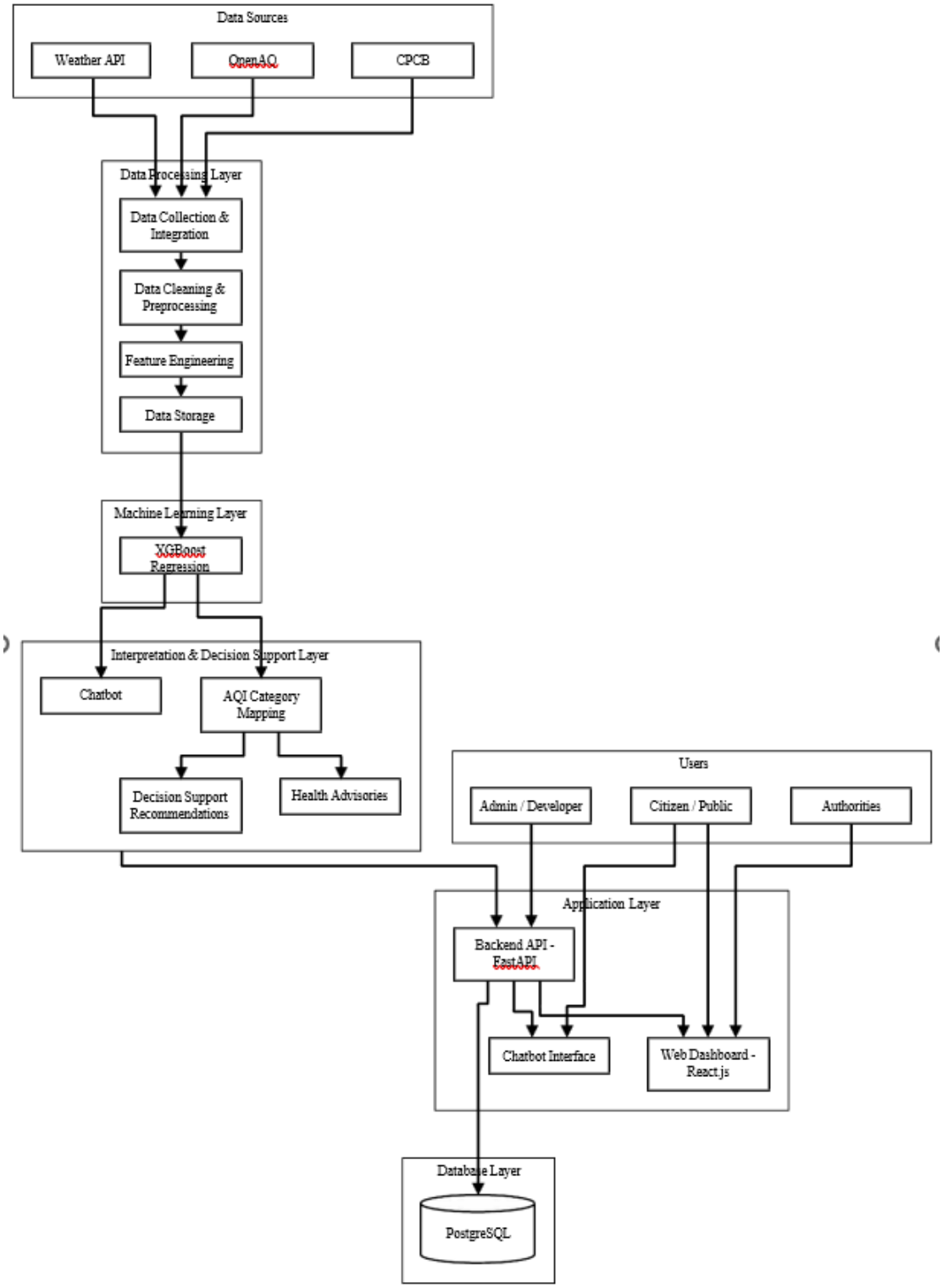


Fig. 1. System architecture of the proposed air-quality forecasting and decision-support dashboard.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

J. Flowchart

Fig. 2 presents the same process as a flowchart. The only decision point is the validation check: if the validation error is not acceptable, the hyperparameters are adjusted and the model is trained again before testing and forecasting.

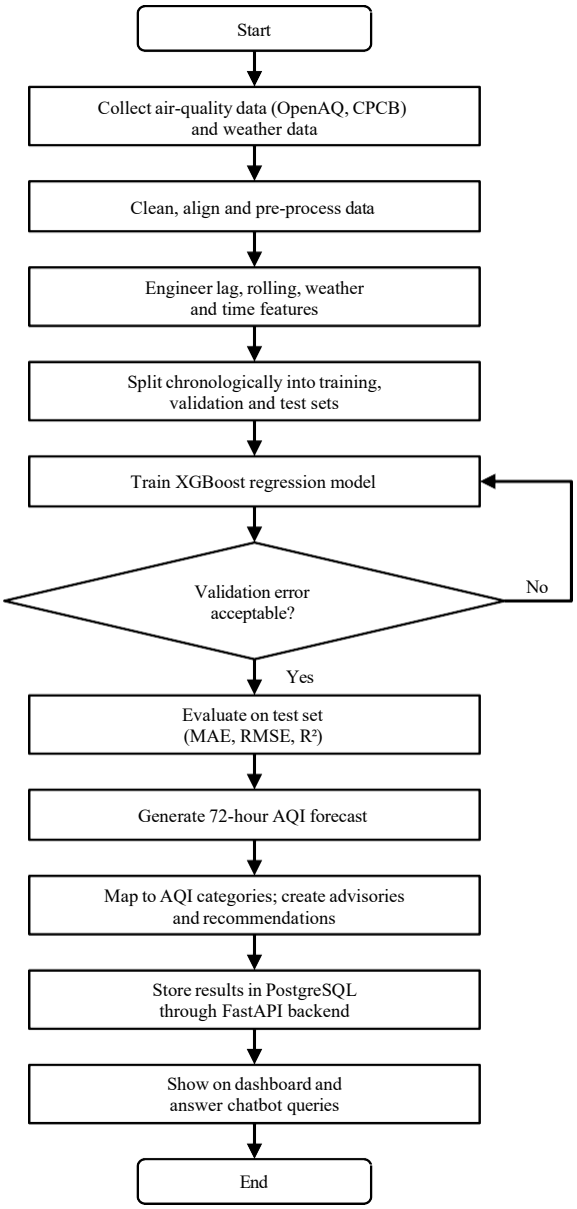


Fig. 2. Flowchart of the proposed AQI forecasting and decision-support process.

IV. DATASET DESCRIPTION

The data set will be assembled from observations of selected monitoring locations, merged with weather parameters on a common time axis. Its final size and composition depend on how complete and compatible the retrieved records turn out to be.

A. Data Sources

Pollutant measurements will be taken from OpenAQ [5] and CPCB [6], subject to availability for the chosen stations. Temperature, relative humidity, wind speed and precipitation will be fetched from a weather source such as the Open-Meteo API [7]. Only records that are compatible across sources will be kept for model building.

B. Dataset Characteristics

The data are expected to fall into four groups, summarised in Table I.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

TABLE I EXPECTED DATA CATEGORIES

Category	Examples	Role
Air quality	PM2.5, PM10, NO_2 and other reported pollutants	Predictor
Meteorological	Temperature, humidity, wind speed, precipitation	Predictor
Temporal	Date, hour, day of week	Predictor
AQI	Reported or computed AQI	Target

AQI values will be obtained from a reliable source or calculated with the applicable AQI method whenever the necessary pollutant concentrations exist. The ultimate set of features depends on data availability and quality.

C. *Data Pre-processing*

Before training, duplicate rows are removed, missing values are handled with suitable techniques, timestamps are standardised and air-quality and weather records are aligned by time and location. Records with invalid values, inconsistent units or unsuitable quality are screened out, and time-based and weather-derived features are prepared. These steps are intended to raise data quality and limit forecasting error.

D. *Dataset Partitioning*

Observations are split chronologically: the earliest portion trains the model, the next portion supports validation and model selection, and the most recent portion is reserved for the final test. A time-ordered split is preferred because air-quality data are sequential; shuffling would let future observations influence training and would overstate performance on unseen periods. Exact partition sizes will be decided after the data have been collected and examined.

V. EXPERIMENTAL SETUP

The system is implemented as software that integrates data acquisition, pre-processing, learning, storage and visualisation. The set-up is meant to measure how well the XGBoost model forecasts AQI and how usefully the dashboard presents the results.

A. *Software and Technologies*

TABLE II TECHNOLOGY STACK

Component	Technology	Purpose
Language	Python	Pre-processing, modelling, backend logic
Model	XGBoost Regression	AQI forecasting
Backend	FastAPI	API, model integration
Frontend	React.js	Interactive dashboard
Database	PostgreSQL	Observations, forecasts, records
Data	OpenAQ, CPCB, Open-Meteo	Air-quality and weather inputs

B. *Model Development*

XGBoost is the principal forecasting algorithm [3]. It is trained on the prepared data containing pollutant readings, weather attributes and time features, with the AQI at the required future time as the target. Support for horizons up to 72 hours is subject to the selected strategy and the available inputs, and hyper-parameters are chosen using the training and validation portions.

C. *Training and Validation Procedure*

The training portion fits the model, the validation portion guides tuning and selection, and the test portion measures performance on data untouched by either step. To avoid leakage, inputs, targets and forecast horizon are aligned so that nothing from a later time is used to predict an earlier AQI value.

D. *System Integration*

The trained model is wrapped by the FastAPI backend, which prepares inputs, generates forecasts and returns them to the React.js dashboard; PostgreSQL stores the observations and prediction records. The chatbot is designed around the same stored data and a set of predefined response rules.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

E. Evaluation Metrics

Forecast accuracy will be quantified with three standard regression measures, where y_i is the observed AQI, $\hat{y}_i$ the predicted AQI, $\bar{y}$ the mean observed AQI and n the number of test samples:

$$MAE = (1/n)\ \Sigma\ |y_i - \hat{y}_i|$$

$$RMSE = \sqrt{[(1/n)\ \Sigma\ (y_i - \hat{y}_i)^2]}\quad R^2 = 1 - \Sigma\ (y_i - \hat{y}_i)^2\ /\ \Sigma\ (y_i - \bar{y})^2$$

Lower MAE and RMSE mean smaller errors, whereas an R^2 closer to 1 indicates that more of the variation in AQI is explained. The measured values will be reported once dataset preparation, training and testing are complete.

VI. EVALUATION PLAN AND EXPECTED OUTCOMES

Because the platform is still under construction, the points below describe how it will be assessed and what it is intended to deliver; they are not results.

A. Forecasting Performance

Predicted AQI values will be compared with observed values in the test period using MAE, RMSE and R^2. Errors will also be examined by forecast horizon and by pollution level to show where the model is more or less reliable.

B. Forecast Reliability

Forecasts for up to 72 hours will be checked against actual observations for the matching periods wherever reference data exist. This will reveal how error changes between short and long horizons and between clean and heavily polluted conditions.

C. Dashboard Functionality

Testing will confirm that the dashboard displays measurements, weather information, forecasts, trends and pollutant details correctly, and that data retrieval, backend communication, database operations and the handling of missing data work as intended.

D. Health Advisory and Decision Support

The assigned category, health precautions and recommendations will be verified against the applicable AQI interpretation rules. These outputs support awareness and planning and do not replace official public-health guidance.

E. Chatbot Evaluation

The chatbot will be tested with questions on AQI, pollutants, forecasts, weather and precautions, and its answers will be judged for relevance, correctness and consistency with system data. It should state clearly when information is unavailable instead of producing unsupported replies.

F. Expected Outcomes

The work is expected to yield: (a) an XGBoost model for AQI forecasting from air-quality and weather data; (b) forecasts up to 72 hours, depending on data and performance; (c) an interactive dashboard with forecasts, trends and pollutant details; (d) rule-based AQI categories and health advisories; (e) a project-specific chatbot; (f) a decision-support interface for citizens and authorities; and (g) an evaluation based on MAE, RMSE and R^2 together with functional testing. Quantitative conclusions will be reported only after the experiments are finished.

VII. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper presents an AI-driven air quality forecasting and smart decision support system that aims to predict the Air Quality Index (AQI) for up to 72 hours using XGBoost regression, air pollution measurements, and weather data. The proposed platform integrates data preprocessing, forecasting, AQI classification, health advisories, and a chatbot into a single web-based dashboard developed using React.js, FastAPI, and PostgreSQL. It is designed to help citizens understand upcoming air quality conditions and support authorities in making informed environmental decisions.

The proposed system has the potential to improve air quality awareness and encourage preventive actions through timely forecasts and easy-to-understand recommendations. Its performance will be evaluated using MAE, RMSE, and R^2 metrics, along with functional testing of the dashboard and chatbot. As the system is currently under development, its actual forecasting accuracy remains to be established through experimental evaluation. Future enhancements may include comparing different machine learning models, expanding coverage to additional cities, integrating real-time alerts, and incorporating explainable AI techniques to improve the transparency and reliability of predictions.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131057

B. Future Scope

Several extensions are possible: comparing other machine-learning and deep-learning models; adding more monitoring stations and weather sources; covering additional cities; improving the chatbot with live data and richer language understanding; studying pollution sources with verified datasets such as fire-hotspot records; sending e-mail, mobile or web alerts when AQI crosses set thresholds; adding explainable-AI techniques to show which factors drive a prediction; and retraining the model periodically on new observations to follow changing pollution patterns.

ACKNOWLEDGMENT

The authors thank the Department of Computer Engineering, Pune District Education Association's College of Engineering, Pune, for the guidance and facilities provided during this work.

REFERENCES

[1]. S. O. Charjan, K. Gajbhiye, J. Warhade, and S. Wankhade, "A feature engineering–driven ensemble approach for accurate AQI forecasting,"*Discover Applied Sciences*, vol. 8, Art. no. 216, 2026, doi: 10.1007/s42452-025-08165-w.

[2]. G. Ravindiran et al., "Impact of air pollutants on climate change and prediction of air quality index using machine learning models,"*Environmental Research*, vol. 239, part 1, Art. no. 117354, 2023, doi: 10.1016/j.envres.2023.117354.

[3]. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[4]. E. Mohapatra, M. Das, and S. Rath, "Deep learning-based AQI forecasting: A CNN-LSTM model with visual insights from SHAP-LIME and PDP," *Discover Applied Sciences*, vol. 7, Art. no. 1326, 2025, doi: 10.1007/s42452-025-07845-x.

[5]. OpenAQ, "Measurements: OpenAQ API documentation." [Online]. Available: https://docs.openaq.org/resources/measurements

[6]. Central Pollution Control Board, "National Air Quality Index." [Online]. Available: https://airquality.cpcb.gov.in/AQI_India/

[7]. Open-Meteo, "Historical weather API documentation." [Online]. Available: https://open-meteo.com/en/docs/historical-weather-api

[8]. Machine learning algorithms to forecast air quality: a survey Related work and comparison of forecasting approaches M. Méndez, M. G. Merayo, and M. Núñez, "Machine learning algorithms to forecast air quality: a survey," Artificial Intelligence Review, vol. 56, pp. 10031–10066, 2023. DOI: 10.1007/s10462-023-10424-4

[9]. A systematic survey of air quality prediction based on deep learning Alternative models and future performance comparisons Z. Zhang, S. Zhang, C. Chen, and J. Yuan, "A systematic survey of air quality prediction based on deep learning," Alexandria Engineering Journal, vol. 93,pp. 128–141, 2024. DOI: 10.1016/j.aej.2024.03.031.

[10]. Interpretable Machine Learning Approaches for Forecasting and Predicting Air Pollution: A Systematic Review Explainable AI and interpretation of model predictions A. Houdou et al., "Interpretable Machine Learning Approaches for Forecasting and Predicting Air Pollution: A Systematic Review," Aerosol and Air Quality Research, vol. 24, Art. no. 230151, 2024. DOI: 10.4209/aaqr.230151

[11]. S. Tırınk, "Machine learning-based forecasting of air quality index under long-term environmental patterns: A comparative approach with XGBoost, LightGBM, and SVM," *PLOS ONE*, vol. 20, no. 10, Art. no. e0334252, 2025. doi: 10.1371/journal.pone.0334252.