iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75622

EcoCast: Predictive Modeling of Urban Air Quality Indices Using Spatiotemporal

Nagaveni[1], Shilpa Joshi[2]
Department of computer Science and Engineering (MCA), Visvesvaraya technological University, CPGS, Kalaburagi

Abstract*: Air pollution has emerged as a critical environmental challenge affecting public health and ecosystem sustainability worldwide. The Air Quality Index serves as a standardized metric for communicating air pollution levels to the general public. This research presents a comprehensive web-based application for predicting AQI values using machine learning techniques, specifically employing Linear Regression algorithms. The system accepts seven key pollutant parameters including ammonia, particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide, and carbon monoxide as input features. The model was trained on historical air quality data and achieved promising prediction accuracy. The implementation utilizes Python Flask framework for backend processing, scikit-learn for machine learning operations, and responsive HTML/CSS/JavaScript for frontend development. The application provides real-time AQI predictions categorized into six quality levels ranging from Good to Severe, enabling users to make informed decisions regarding outdoor activities and health precautions. Performance evaluation demonstrates the model's capability to accurately predict AQI values with minimal error margins. This system offers a practical tool for environmental monitoring agencies, healthcare professionals, and general public to assess air quality conditions effectively.

Keywords: Air Quality Index, Machine Learning, Linear Regression, Pollutant Prediction, Web Application, Environmental Monitoring, Flask Framework, Real-time Prediction

I. INTRODUCTION

Air quality has become a paramount concern in contemporary society, particularly in rapidly urbanizing regions where industrial activities, vehicular emissions, and various anthropogenic factors contribute to atmospheric pollution. The degradation of air quality poses severe threats to human health, causing respiratory diseases, cardiovascular complications, and reduced life expectancy. According to environmental health organizations, millions of premature deaths occur annually due to exposure to polluted air, making air quality monitoring and prediction essential for public welfare.

The Air Quality Index serves as a standardized measurement system designed to quantify and communicate the level of air pollution to the general public in an easily understandable format. AQI calculations incorporate multiple pollutant concentrations including particulate matter, gaseous pollutants, and other harmful substances present in the atmosphere. Traditional air quality assessment methods rely on physical monitoring stations that provide historical data but lack predictive capabilities for future conditions.

Machine learning technologies have revolutionized environmental monitoring by enabling accurate predictions based on historical data patterns. These computational approaches can identify complex relationships between multiple pollutant parameters and generate reliable forecasts of air quality conditions. Linear regression, despite being a fundamental machine learning algorithm, demonstrates remarkable effectiveness in modeling relationships between independent variables (pollutant concentrations) and dependent variables (AQI values).

This research addresses the critical need for accessible, real-time air quality prediction tools by developing a comprehensive web-based application. The system integrates machine learning algorithms with modern web technologies to create a user-friendly platform for AQI prediction. The application accepts seven primary pollutant measurements as input parameters and generates accurate AQI predictions with corresponding quality categories, enabling stakeholders to take appropriate preventive measures.

The primary objectives of this research include: developing a robust machine learning model for AQI prediction, creating an intuitive web interface for user interaction, implementing real-time prediction capabilities with categorical classification, and evaluating the model's performance through comprehensive testing. The system architecture emphasizes scalability, reliability, and ease of deployment, making it suitable for integration with existing environmental monitoring infrastructure.

II. LITERATURE REVIEW

Numerous researchers have investigated air quality prediction using various machine learning approaches. Kumar and colleagues explored the application of multiple regression techniques for AQI forecasting in urban environments, demonstrating that ensemble methods could achieve superior accuracy compared to individual algorithms. Their study highlighted the importance of feature selection and data preprocessing in improving prediction performance.

Zhang et al. conducted comprehensive research on deep learning architectures for air quality prediction, implementing recurrent neural networks and long short-term memory models. Their findings indicated that temporal dependencies in air quality data could be effectively captured using sequential learning approaches. However, they noted that simpler models like linear regression often provided comparable results for short-term predictions while requiring significantly less computational resources.

Environmental monitoring agencies have increasingly adopted machine learning technologies for operational forecasting systems. The European Environment Agency published guidelines recommending the integration of predictive models with real-time monitoring networks to enhance early warning capabilities. These recommendations emphasized the need for interpretable models that domain experts could validate and trust.

Research on web-based environmental monitoring systems has demonstrated the effectiveness of combining backend machine learning services with interactive frontend interfaces. Patel and team developed a cloud-based platform for multi-city air quality monitoring, incorporating RESTful APIs for seamless data exchange between components. Their architecture inspired many subsequent implementations, including the approach adopted in this research.

Studies comparing different machine learning algorithms for AQI prediction have yielded varied conclusions depending on data characteristics and prediction horizons. Linear regression models have been shown to perform exceptionally well for datasets with strong linear relationships between pollutant concentrations and AQI values. The simplicity and interpretability of linear models make them particularly suitable for operational deployment where model transparency is essential.

III. METHODOLOGY

A. System Architecture

The proposed system follows a three-tier architecture comprising presentation layer, application layer, and data layer. The presentation layer consists of responsive HTML interfaces styled with CSS and enhanced with JavaScript for dynamic interactions. The application layer implements Flask framework for request handling, routing, and business logic execution. The data layer manages the trained machine learning model stored in pickle format for efficient loading and inference.

B. Dataset Description

The training dataset comprises historical air quality measurements collected from environmental monitoring stations. Each record contains seven pollutant concentration values: ammonia (NH_3), particulate matter with diameter less than 2.5 micrometers (PM2.5), particulate matter with diameter less than 10 micrometers (PM10), ozone (O_3), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), and carbon monoxide (CO). The corresponding AQI values serve as target labels for supervised learning. Data preprocessing included handling missing values, outlier detection, and feature normalization to ensure optimal model training.

C. Machine Learning Model

1) Linear Regression Algorithm:

Linear regression establishes a linear relationship between input features and output variable through the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_7 x_7 + \varepsilon$$

where y represents the predicted AQI value, x_1 through x_7 denote the seven pollutant concentrations, β_0 is the intercept term, β_1 through β_7 are coefficient parameters, and ε represents the error term. The model training process minimizes the sum of squared residuals between predicted and actual AQI values using ordinary least squares optimization.

2) Model Training:

The dataset was partitioned into training and testing subsets using a 70:30 split ratio, with random state parameter set to zero for reproducibility. The scikit-learn LinearRegression class was instantiated and fitted on the training data.

Model parameters were learned through iterative optimization, and the trained model was serialized using Python's pickle module for persistent storage and rapid loading during inference.

D. Web Application Development

1) Backend Implementation:

The Flask application defines routes for handling GET and POST requests. The root endpoint serves the HTML interface for GET requests and processes prediction requests for POST methods. Input validation ensures that all seven pollutant parameters are provided and can be converted to floating-point numbers. The loaded model performs inference on the input array, and results are returned as JSON responses containing the predicted AQI value.

2) Frontend Design:

The user interface employs modern web design principles with responsive layouts adapting to various screen sizes. Input forms collect pollutant concentration values through labeled text fields with appropriate placeholders. JavaScript event handlers intercept form submissions, execute asynchronous fetch requests to the backend API, and dynamically update the DOM to display prediction results without page reloads. CSS styling implements visual hierarchies, color-coded AQI categories, and smooth transitions for enhanced user experience.

E. AQI Classification System

The predicted numerical AQI values are categorized into six qualitative levels based on internationally recognized standards. Values between 0 and 50 indicate Good air quality with minimal health impact. The range 51 to 100 represents Satisfactory conditions suitable for most individuals. Moderately Polluted air falls within 101 to 150, potentially affecting sensitive groups. Poor air quality spans 151 to 200, causing discomfort to many people. Very Poor conditions between 201 and 300 necessitate health advisories. Severe pollution above 300 requires immediate protective actions and activity restrictions.

IV. IMPLEMENTATION

A. Development Environment

The system was developed using Python 3.x with essential libraries including Flask for web framework, NumPy for numerical computations, Pandas for data manipulation, and scikit-learn for machine learning operations. The requirements.txt file specifies all dependencies with version constraints to ensure reproducible deployments. Development utilized integrated development environments with debugging capabilities and version control systems for collaborative development.

B. Model Training Process

The training script (linear.py) loads the dataset from CSV format and extracts feature columns and target variable using Pandas DataFrame slicing operations. The train_test_split function partitions data while maintaining random state for consistent results across multiple executions. The LinearRegression model is instantiated with default parameters and fitted on training data. Prediction accuracy is evaluated on the test set, and the trained model is serialized to disk using pickle.dump() for subsequent loading in the web application.

C. Web Server Configuration

The Flask application initializes by loading the pre-trained model from the pickle file during server startup. Route decorators define URL patterns and accepted HTTP methods for each endpoint. The predict_aqi function implements request handling logic, extracting form data, constructing input arrays, invoking model prediction, and formatting JSON responses. Exception handling mechanisms catch and report errors gracefully, preventing server crashes from invalid inputs.

D. User Interface Components

The frontend comprises multiple HTML pages with consistent navigation structures. The main prediction interface contains a form with seven input fields corresponding to pollutant parameters. JavaScript validation ensures that users provide numeric values within acceptable ranges before submission. The results section displays predicted AQI values with color-coded backgrounds matching the quality category. Additional informational pages explain AQI meanings, health implications, and recommended actions for different pollution levels.

TABLE I
POLLUTANT PARAMETERS AND MEASUREMENT UNITS

Parameter	Chemical Formula	Unit	Typical Range
Ammonia	NH_3	µg/m³	0-400
PM2.5	-	µg/m³	0-500
PM10	-	µg/m³	0-600
Ozone	O_3	µg/m³	0-300
Nitrogen Dioxide	NO_2	µg/m³	0-200
Sulfur Dioxide	SO_2	µg/m³	0-80
Carbon Monoxide	CO	mg/m³	0-40

V. RESULTS AND DISCUSSION

A. Model Performance Evaluation

The trained linear regression model demonstrated satisfactory performance on the test dataset. Evaluation metrics including Mean Absolute Error, Root Mean Squared Error, and R-squared coefficient indicate that the model captures the underlying relationship between pollutant concentrations and AQI values effectively. The R-squared value approaching unity suggests that the majority of variance in AQI measurements is explained by the input features, validating the appropriateness of linear regression for this application.

B. Prediction Accuracy

Analysis of prediction errors reveals that the model achieves highest accuracy for moderate AQI ranges while showing slightly increased errors for extreme pollution conditions. This behavior is expected given the distribution of training data, which predominantly contains observations from moderate pollution levels. The model successfully identifies the correct AQI category for the majority of test cases, ensuring that users receive appropriate health advisories even when absolute numerical predictions deviate from true values.

TABLE II
AQI CATEGORY CLASSIFICATION

AQI Range	Category	Color Code	Health Implications
0-50	Good	Green	Minimal impact
51-100	Satisfactory	Yellow	Minor breathing discomfort to sensitive people
101-150	Moderately Polluted	Orange	Breathing discomfort to people with lung disease
151-200	Poor	Red	Breathing discomfort to most people
201-300	Very Poor	Purple	Respiratory illness to people on prolonged exposure
301-500	Severe	Maroon	Affects healthy people and seriously impacts those with existing diseases

C. Web Application Performance

The deployed web application exhibits responsive behavior with minimal latency between user input submission and result display. Server-side processing time remains consistently low due to the computational efficiency of linear regression inference.

The Flask application handles concurrent requests effectively, making the system suitable for deployment in production environments with moderate user loads. Browser compatibility testing confirms proper functionality across major web browsers including Chrome, Firefox, Safari, and Edge.

D. User Experience Analysis

Informal user testing revealed positive feedback regarding interface intuitiveness and result presentation clarity. Users appreciated the color-coded category display and detailed health implications associated with each AQI level. Some users suggested additional features such as historical trend visualization and location-based predictions, which represent promising directions for future enhancements. The responsive design adapts seamlessly to mobile devices, enabling air quality checks on smartphones and tablets.

E. Comparative Analysis

Comparing the linear regression approach with alternative machine learning algorithms reveals trade-offs between complexity and performance. While advanced methods such as random forests or neural networks might achieve marginally better accuracy, linear regression offers advantages in terms of model interpretability, training speed, and computational resource requirements. For practical deployment scenarios where rapid predictions and model transparency are priorities, linear regression presents an optimal balance of simplicity and effectiveness.

VI. CONCLUSION

This research successfully developed and implemented a comprehensive web-based system for Air Quality Index prediction using machine learning techniques. The linear regression model demonstrates reliable performance in forecasting AQI values based on seven key pollutant parameters, providing accurate predictions suitable for real-world applications. The integration of Flask framework for backend processing and responsive web technologies for frontend presentation creates a user-friendly platform accessible to diverse stakeholders including environmental agencies, healthcare professionals, and general public.

The system addresses the critical need for accessible air quality prediction tools by offering real-time AQI forecasts with categorical classifications aligned with international standards. Performance evaluation confirms the model's capability to accurately predict air quality conditions across various pollution levels, enabling users to make informed decisions regarding outdoor activities and health precautions. The modular architecture facilitates future enhancements and integration with additional data sources or predictive algorithms.

Future research directions include incorporating temporal dependencies through time series models, expanding the feature set to include meteorological parameters, implementing ensemble learning approaches for improved accuracy, and developing mobile applications for enhanced accessibility. The integration of real-time data feeds from monitoring stations would enable continuous prediction updates, transforming the system into a comprehensive air quality forecasting platform. Additionally, explainable AI techniques could be incorporated to provide users with insights into which pollutants contribute most significantly to predicted AQI values.

The successful implementation of this system demonstrates the practical applicability of machine learning technologies in environmental monitoring and public health protection. By combining rigorous scientific methodologies with accessible technology platforms, this research contributes to the broader objective of creating healthier, more sustainable urban environments through data-driven decision-making and proactive environmental management.

VII. ACKNOWLEDGMENT

The authors would like to express sincere gratitude to the environmental monitoring agencies for providing access to air quality datasets used in this research. Special thanks to the university faculty members who provided valuable guidance throughout the project development. We acknowledge the open-source community for developing and maintaining the software libraries that made this implementation possible, including Flask, scikit-learn, NumPy, and Pandas.

REFERENCES

[1] A. Kumar, P. Goyal, and A. K. Gupta, "Air quality forecasting using advanced machine learning techniques: A review," *Atmospheric Environment*, vol. 235, pp. 117-129, 2020.
[2] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, "Cautionary tales on air-quality improvement in Beijing," *Proceedings of the Royal Society A*, vol. 473, no. 2205, pp. 20170457, 2017.

[3] J. Ma, J. C. Cheng, C. Jiang, W. Chen, and R. Zhang, "Real-time prediction of air quality index based on LSTM neural network," in *Proc. International Conference on Smart Infrastructure and Construction*, Cambridge, UK, 2019, pp. 255-262.
[4] R. Patel, N. Shah, and M. Desai, "Cloud-based environmental monitoring system with machine learning integration," *IEEE Access*, vol. 8, pp. 145623-145635, 2020.
[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
[6] World Health Organization, "WHO global air quality guidelines: Particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide," Geneva, Switzerland, 2021.
[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2009.
[8] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.
[9] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
[10] European Environment Agency, "Air quality in Europe — 2020 report," EEA Report No. 09/2020, Copenhagen, Denmark, 2020.
[11] S. Rao, "Development of web-based air quality monitoring and forecasting system using machine learning," M.S. thesis, Dept. Computer Science, Stanford University, Stanford, CA, 2019.
[12] Central Pollution Control Board, "National Air Quality Index," Ministry of Environment, Forest and Climate Change, Government of India, New Delhi, India, Tech. Rep. AQI-2014, 2014.