International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 19s, 2025
https://theaspd.com/index.php

Ai Based Real-Time Air Quality Monitoring System

Shobha T[1], Sreevidya B S[2], Shreya B[3], Surya P[4], Vaishnavi B Huvannavar[5]
[1,2] Computer Science and Engineering, B.M.S College of Engineering, Bengaluru, Karnataka, India
[3,4,5] Information Science and Engineering, B.M.S College of Engineering, Bengaluru, Karnataka, India
shobha.ise@bmsce.ac.in[1], sreevidyabs.ise@bmsce.ac.in[2], shreya.is21@bmsce.ac.in[3], surya.is21@bmsce.ac.in[4], vaishnavi.is21@bmsce.ac.in[5]

Abstract:
Air pollution has become one of the most critical environmental and public health issues globally, affecting quality of life and contributing to various health problems. With increasing urbanization and industrialization, the need for effective air quality monitoring has never been more urgent. This study presents a Real-Time Air Quality Monitoring System that delivers real-time air quality data, predictive analytics, and health recommendations. The system integrates continuous data collection from monitoring stations, a Bi-Directional Long Short-Term Memory (Bi-LSTM) model for air quality forecasting, and a visualization dashboard for data interpretation.
To enhance predictive accuracy, Particle Swarm Optimization (PSO) is used to optimize model performance, while a generative AI-based chatbot improves user interaction by offering personalized insights and preventive recommendations. Experimental results confirm the system's effectiveness in forecasting pollution trends, identifying hotspots, and supporting informed decision-making for both individuals and policymakers. Additionally, an agent-based simulation module enables policymakers to evaluate the impact of interventions on air quality under various real-world scenarios. This research contributes to environmental monitoring and public health management, providing a scalable, intelligent approach to mitigating air pollution.
Keywords: *Real-Time Air Quality Monitoring System, BiLSTM, Particle Swarm Optimization, Environmental Monitoring, Deep Learning, NLP Chatbot, agent-based simulation*

I. INTRODUCTION

Air pollution has become a critical environmental issue, with significant implications for public health, quality of life, and the environment. Monitoring and managing air quality are essential for understanding pollution levels and protecting communities from their harmful effects. Real-time measurements and analysis are necessary to properly monitor pollutant concentrations, identify hotspots of high pollution levels, and provide forecasting details regarding upcoming air quality trends. Certain key indications of air pollution are visible in cities such as Delhi, with the rapid growth of urban zones, industrialization, and exhaust from motor vehicles causing air conditions to deteriorate critical pollutants such as PM2.5, PM10, NO, CO, and O, inflict critical health risks in the guises of respiratory and cardiovascular diseases. Even though the technological sphere remains undefeated, India has some of the world's most polluted cities which is alarming. The National Capital Region (NCR) is well-known for having excelled infinitely at PM2.5, surpassing the World Health Organization standards. Delhi constantly remains at the top of the leader board as the dirtiest city to live in, along with other neighbouring states of Haryana, Uttar Pradesh, and Bihar.

The remainder is said to be history. Despite India's great efforts to battling pollution, they surprisingly manage to maintain air quality in urbanized cities. On the other side of the globe, Bangalore is known as the Silicon Valley of India boasting pleasant weather to jog outdoors, but continues to face pointers of shifting into the 'not so clean' list. In addition to the NO and CO gases emitted from traffic congestion, dust and particulate matter emitted during construction and industrial growth further lower air quality. Reports state that Bangalore ranks among the top cities suffering from PM2.5 levels exceeding safe ranges. Recent data show that the city stands at a staggering 10th position in the country's list of most polluted cities and is especially notorious for its peak hour traffic.

Despite these efforts, India has adopted a traditional approach to monitoring air quality which suffers from a lack of real-time forecasting capabilities. Access is limited to a small set of policymakers and the

general public without predictive analytics software, highlighting the necessity for proactive interventions to reduce pollution.
To meet such needs, this research discusses an AI-based Air Quality Monitoring and Prediction System that combines real-time pollutant information from 14 Continuous Ambient Air Quality Monitoring Stations (CAAQMS) spread across Bangalore.

II.LITERATURE SURVEY

Air pollution has become a pressing global concern, driving extensive research on air quality prediction using advanced machine learning (ML) and deep learning (DL) techniques.

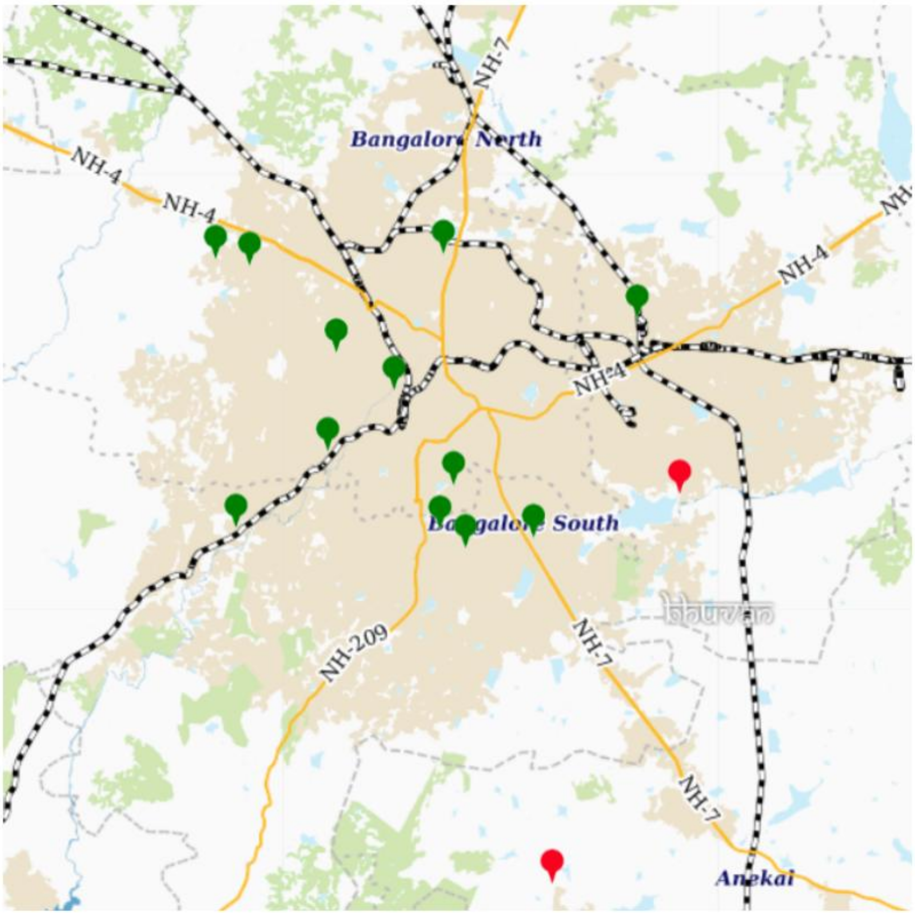

Fig. 1: Monitoring stations of Bangalore from CPCB website. (Green pins represent live stations and red pin represent inactive stations).

Researchers worldwide have explored a various approach, from traditional methods to sophisticated neural networks, to improve forecasting accuracy and facilitate proactive environmental management.
Fathima et al. [1] emphasized the efficacy of deep learning models in air quality forecasting and, proposed the capability of neural networks to effectively identify temporal trends in pollution data. Rahmat et al. [2] complemented this with an extension of including LSTM networks to provide safe levels of air quality for human activity, proposing its capability in adaptive environmental planning. Furthermore, Sur et al. [3] used ML models to forecast the level of pollution in Delhi and hotspot detection as a follow-up to Bhatia et al. [5], who used hotspot detection and forecasting, as they introduced the use of spatial data in air quality analysis.
The use of LSTM networks for air quality forecasting has also been reported in other studies. Thaweephol et al. [4] utilized the application of an LSTM model in Bangkok PM2.5 to establish its ability to model complicated sequential data. Their work agrees with that of Chaudhary, et al. [7], who focused on the LSTM-based prediction of Indian megacity air pollutant concentration time series. Samal et al. [8] further

applied the deep learning-Kriging interpolation combination to ascertain the application of spatio-temporal modeling to improve predictability. Cao et al. [9] proposed enhanced by proposing a hybrid model based on empirical mode decomposition for removal of pollution data with improved accuracy. Further deep-learning-based forecast innovation was proposed by Wei et al. [10], who proposed an LSTM-based air pollution forecasting framework, and Pandey et al. [11], who applied a bidirectional LSTM (BiDLSTM) network to improve the power of predictability using past as well as future associations. Samal et al. [12] further focused on spatial consideration using distance-based interpolation combined with deep learning and proposed an integrated approach for air quality forecasting.

Dua et al. [13] proposed an attention-based bidirectional LSTM network for real-time air pollution prediction that is more accurate and interpretable. Their method was similar to that of Grace et al. [15], who maximized the performance of Bi-LSTM with Spider Monkey Optimization for precise air quality index estimation. Faiz et al. [14] demonstrated how artificial intelligence techniques can be applied to urban pollution forecasting in Chennai, highlighting the practical applications of AI-driven environmental monitoring.

Parallel to air quality prediction, advancements in AI-driven chatbot systems have been explored for information retrieval and automation. Shetage et al. [16] developed an NLP-powered chatbot for automated FAQ handling, leveraging ML to enhance the response accuracy. Their work complements Singidi SugunaSri et al. [17], who proposed a chatbot integrated question-answering system, and Kale et al. [18], who introduced FAQ-Gen, an automated system for domain specific FAQ generation.

Khennouche et al. [19] analyzed the challenges of deploying generative AI chatbots, such as ChatGPT, in FAQ automation, addressing concerns regarding response accuracy and user experience. Euodia Louis [20] further examined user interactions with conversational AI, shedding light on engagement patterns and optimization strategies for chatbot usability. These studies provide valuable insights for improving AI-powered information retrieval systems, which could complement air quality prediction models by offering real-time pollution insights through interactive chatbot interfaces.

This extensive review of the literature highlights advances in air quality forecasting using ML and DL techniques while also highlighting the expanding role of AI-powered chatbot systems in information retrieval. Integration of these technologies can provide more effective, intelligent and effective environmental monitoring solutions.

III. DATA PREPARATION AND PRE-PROCESSING

Effective data preprocessing is required to ensure the quality and reliability of any data-driven model, especially in environmental datasets where inconsistencies such as missing irregular spacing, noise, and values are common. This section explains the scientific protocols adopted in preparing the air quality dataset employed in this study for exploratory and predictive analyses.

A. Data Collection and Initial Exploration

The data were collected from a publicly available repository containing air quality measurements for some Indian cities. Python packages such as texttt{Pandas} and texttt{NumPy} were utilized for data loading and preliminary exploration. Summary statistics were computed for distribution inspection, anomaly detection, and verification of completeness of data Key variables included the observation date, city of measurement, pollutant concentrations ($PM_{2.5}$, PM_{10}, NO_2, SO_2, CO, O_3), and the computed Air Quality Index (AQI).

B. Handling Missing Values

Missing data, a common issue in environmental monitoring, were addressed using multiple strategies. Where feasible, missing pollutant values were imputed using the mean or median within the city or date groups. Records with critical pollutant values or missing AQI values were removed. Time series continuity was maintained using forward and backward fill techniques when applicable.

C. Date and Time Feature Engineering

The Date column was converted to a date time format to facilitate temporal operations. Additional temporal features such as year, month, day of the week, and seasonal indicators were derived. These features are essential for capturing time-based trends in pollution levels.

D. Duplicate and Irrelevant Data Removal
Redundant records were identified and eliminated to maintain data integrity. Non-informative or unrelated fields were dropped to streamline the dataset and reduce the model complexity.
E. Feature Engineering
Several new features were created to improve model performance and interpretability, these included pollutant ratios such as $PM_{2.5}/PM_{10}$ to assess particulate composition, categorical AQI levels based on CPCB guidelines ("Good," "Moderate," "Unhealthy," etc.), and city-level pollutant aggregates computed on a monthly basis to reveal spatial and temporal pollution patterns.
F. Data Transformation and Scaling
Although some machine learning models such as decision trees, are scale-invariant models, such as Support Vector Machines and k-nearest neighbours, require standardized input. Hence, the numerical features were scaled using Z-score normalization or min-max scaling. Logarithmic transformation was applied to skewed variables like $PM_{2.5}$ and CO to reduce the influence of outliers.
G. Outlier Detection and Treatment
Outliers were identified using statistical methods including Interquartile Range (IQR) and Z-score techniques. Domain knowledge is employed to set realistic pollutant thresholds. Depending on the context, outliers were either capped (winsorized) or removed, particularly if deemed erroneous (e.g., negative concentrations).
H. Final Dataset Consolidation
The cleaned and transformed datasets were consolidated into a structured format suitable for both exploratory analysis and model training. The final version was exported in CSV format and archived to ensure reproducibility of the results.

IV. PROPOSED SOLUTION
A. Automated Data Extraction
Air quality data were programmatically extracted from the Central Pollution Control Board (CPCB) website using the Selenium WebDriver. Pollutant-specific metrics including $PM_{2.5}$, PM_{10}, NO_2, CO, SO_2, and O_3 were scraped, parsed, and transformed into a structured format. This automation eliminates the need for manual data entry and ensures that real-time updates are continuously integrated into the system.
B. Data Storage and Management
The collected data were stored in a MongoDB NoSQL database, chosen for its flexibility in handling semi-structured environmental data and its ability to support scalable queries. Each record included pollutant concentrations, station identifiers, and timestamp metadata, enabling both point-in-time querying and temporal trend analysis.
C. AQI Computation and Analysis
Air Quality Index (AQI) is computed following the standardized breakpoints defined by the Central Pollution Control Board (CPCB). For each pollutant, the concentration values were mapped to the corresponding AQI sub-indices using linear interpolation. The final AQI for a location is determined as the maximum of all sub indices and is classified into standard health impact bands such as "Good," "Moderate," "Unhealthy" for Sensitive groups," "Unhealthy", "Very Unhealthy" or "Hazardous."
The implementation supports AQI calculations, extensive data validation, and visualization. Libraries such as plotly are used to depict pollutant distribution, compare station level data, and analyze AQI variations over time. This setup promotes reproducibility and supports further research by enabling easy modifications and experimentation in a collaborative environment.
D. Backend and API Services
A Python-based Flask backend serves as the system integration layer, exposing RESTful API endpoints that provide access to both real-time and historical AQI data. Key functionalities include retrieving AQI values and dominant pollutants for a specified city or station, listing pollutant-specific data and trends, and generating weekly summaries and graphical representations. These APIs enable secure data access via HTTP requests and ensure flexible interactions between the system components.

E. AI-Driven Conversational Interface

To enhance user engagement and make environmental data more accessible, the system incorporates a natural language chatbot interface. Built using the Gemini 2.0 large language model and ChromaDB for context retrieval, the chatbot intelligently responds to user queries. By interfacing with backend APIs, the assistant provides the following insights:
• Real-time AQI levels in specified locations, • Explanations of poor air quality conditions, • Health recommendations based on AQI classifications.
This feature democratizes air quality information and supports non-technical users in making informed decisions.

F. User Interface and Visualization

The web-based user interface was developed using StreamLit, offering a responsive and interactive dashboard for visualizing air quality data. It features an AQI map rendered using the Folium library, where monitoring stations are displayed as color-coded markers based on their AQI levels. Users can choose stations to view AQI values and pollutant concentrations, and access real-time summaries along with relevant health advisories. The interface also provides graphical views of historical trends and comparative analyses to support informed decision-making. Embedded AI assistants offer personalized insights and contextual guidance, further enhancing user interaction.

G. Agent-Based Simulation for Air Quality Modelling

To simulate and analyze the dynamic contribution of different real-world entities to air pollution, we propose an agent-based modelling (ABM) framework implemented using the Mesa Python library. The simulation environment consisted of a 10×10 grid where various vehicles, industrial units, residential blocks, and weather systems—interacted and influenced air quality parameters over time.
Each agent is modelled as a subclass of the agent class in Mesa and assigned a specific type (vehicle, industry, residential, or weather). In each simulation step, agents contributed to changes in pollutant concentrations based on predefined emission rules associated with their type.

H. System Modularity and Extensibility

The proposed system is entirely software-based and operates independently of IoT devices or external sensing hardware. This approach simplifies deployment across diverse geographic regions by eliminating the need for specialized infrastructure. The modular architecture of the system supports future enhancements, including the addition of new pollutant types, integration with satellite-based environmental data, and deployment as a scalable cloud-based service. This extensibility ensures that the system can evolve to meet the expanding requirements and adapt to advancements in environmental monitoring technologies.

I. Predictive Modelling Architecture

This study introduces a hybrid deep learning framework that integrates a Bidirectional Long Short-Term Memory (BiLSTM) network with a transformer-based attention mechanism, which is further enhanced through Particle Swarm Optimization (PSO). The Bi-LSTM component was designed to effectively capture sequential dependencies in pollutant time series data from both past and future contexts. To improve the focus on influential time steps, we incorporate Transformer-style self-attention, allowing the model to assign adaptive importance weights across the sequence.
To optimize model performance, PSO is utilized to automatically search for the best hyper parameters, such as the number of LSTM units, attention dimensions, dropout rate, and learning rate—through a population-based, heuristic search strategy inspired by the social behaviour of swarms. By combining temporal sensitivity, contextual awareness, and adaptive optimization, this approach offers improved predictive accuracy and interpretability for air quality monitoring applications.

V. IMPLEMENTATION

The proposed air quality monitoring system was implemented using a modular architecture comprising data acquisition, AQI computation, API development, a user interface, and intelligent chatbot integration. Each component was designed to operate independently while interacting through well-defined interfaces. The core predictive engine is a Bi-LSTM model enhanced with transformer-based attention and optimized using Particle Swarm Optimization (PSO), which enables accurate and context-aware forecasting of key pollutant levels.

Ethics approval: This study did not involve human participants, human data, or animals, and therefore did not require ethics approval.

A. Data Acquisition and Storage

Air quality data were programmatically extracted from the Central Pollution Control Board (CPCB) website using Selenium WebDriver, which automates browser interaction to download hourly Excel reports. The downloaded files were parsed using the Pandas library to extract pollutant concentration values such as $PM_{2.5}$, PM_{10}, NO_2, SO_2, CO, and O_3. The Pre-processing steps include handling missing values, converting timestamps, and cleaning inconsistent entries. The data are stored for processing in a MongoDB NoSQL database, where each document contains the station name, pollutant concentrations, calculated AQI, and timestamp.

B. AQI Computation

The system adheres to the Air Quality Index (AQI) computation standards laid down by the Central Pollution Control Board (CPCB). The AQI for all pollutants was calculated using a linear interpolation formula based on the CPCB-defined breakpoint ranges. The formula used particularly is as shown in Eq 1:

$$\mathrm{AQI} = \left(\frac{\mathrm{I_{hi}} - \mathrm{I_{lo}}}{\mathrm{BP_{hi}} - \mathrm{BP_{lo}}}\right) \mathrm{X}\ (\mathrm{C} - \mathrm{BP_{lo}}) + \mathrm{I_{lo}} \qquad \text{Eq 1}$$

where C is the measured concentration of the pollutant, BP_{lo} and BP_{hi} are the lower and upper breakpoints off the pollutant, and I_{lo} and I_{hi} are the corresponding index breakpoints. are the corresponding AQI values for those breakpoints. This is performed for each pollutant, generating a set of sub-indices. The overall AQI for a place and time is then determined as the maximum of all valid sub-indices, and the pollutant whose upper sub-index value produces this maximum is computed as the leading cause of air quality loss. To deliver rigor throughout the system, this reasoning is encoded in reusable Python code that is appropriate for backend APIs and data exploration analysis pipelines.

C. Backend API Development

A RESTful API service was developed using the Flask framework to provide air quality data to external consumers. The API includes several endpoints, such as aqi, which returns the latest AQI for all available monitoring stations, and current data, which provides current pollutant concentrations and AQI for a specific station. Each endpoint returns data in the JSON format, enabling seamless integration with frontend interfaces and third-party applications. The API supports query parameters to facilitate filtering by date, location, and pollutant type, thereby offering flexibility in data retrieval and analysis.

D. Web-Based User Interface

A web-based dashboard was implemented using StreamLit to visualize the air quality data interactively. The dashboard features an interactive AQI map created with the Folium library, where station markers are color-coded based on AQI levels. A sidebar allows users to filter data by city, station, and time range, offering a tailored view of air quality metrics. Real-time summaries the pollutant levels are provided along with the corresponding health advisories. In additionally, an embedded chatbot offers contextual guidance that enhance user engagement. The dashboard was optimized for responsiveness and lightweight deployment, ensuring accessibility across both desktop and mobile platforms.

E. Conversational Chatbot Integration

To improve accessibility and public engagement, a natural language chatbot was integrated. The system leverages the Gemini 2.0 large language model API for natural language understanding and response generation, and ChromaDB for context-aware data retrieval. User queries are processed to extract entities such as location and date, which are used to query backend APIs. Responses were dynamically generated with contextual insights based on AQI trends, weather data, and environmental semantics.

F. Exploratory Analysis and Visualization

Exploratory analysis was conducted to understand spatial patterns and air quality dynamics using agent-based simulation and pollutant categorization. The system utilizes real-world AQI data to define risk zones based on pollutant concentration levels and to simulate exposure patterns across these zones. Interactive heatmaps visualize AQI variations and provide spatial insights into pollution severity. The analysis also incorporates user surveys to simulate behavioural responses and validate the perceived air quality. Visual outputs were generated by plotting libraries to present the AQI distributions, category-based health advisories, and region-specific risk summaries. Although timeseries forecasting is not implemented in the current version, the framework is compatible with predictive models, such as ARIMA and Prophet.

VI. RESULTS

The proposed system was evaluated on the basis of real-time data, model accuracy, API performance, and user usability. The following subsections describe the key results.

Fig 2 shows the dashboard of the centralized system

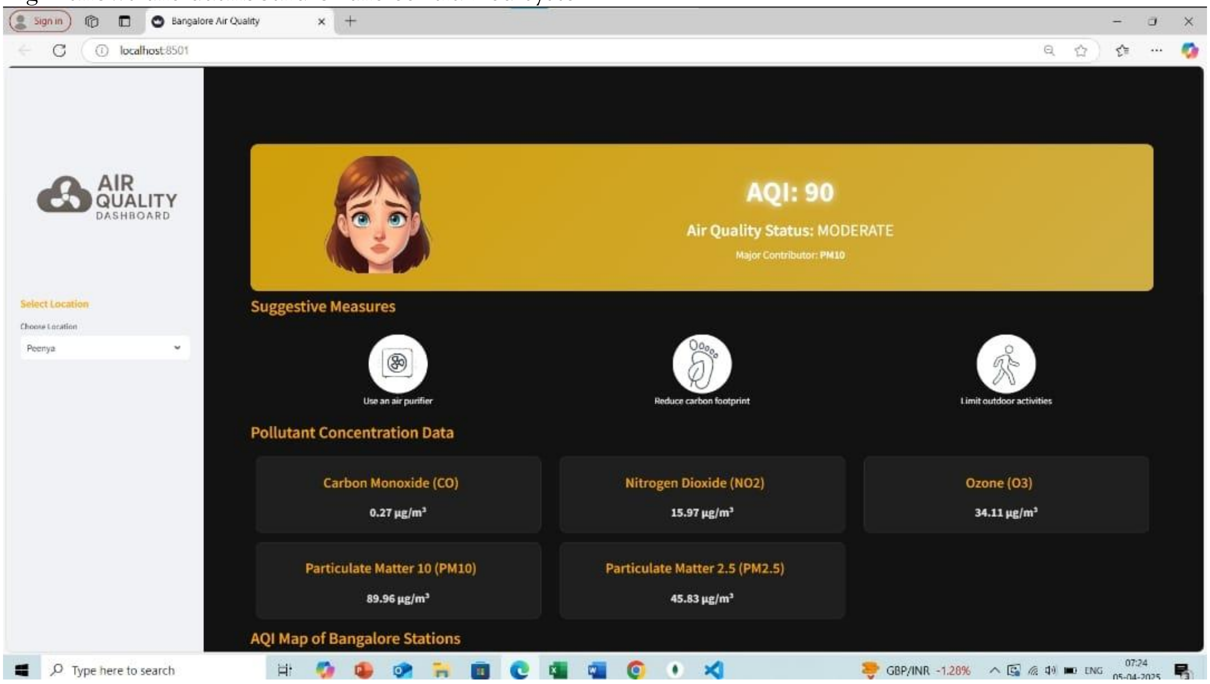

Fig. 2: Dashboard

A. Model Performance

The Bi-LSTM model optimized with Particle Swarm Optimization (PSO) achieved superior performance over baseline models such as Random Forest, standard LSTM and the Bi-LSTM model optimizer. The evaluation was conducted for five target pollutants: PM2.5, PM10, NO, CO, and O.

These results confirm that the model accurately captures temporal pollutant patterns and delivers robust forecasts that are suitable for real-time use.

B. AQI Computation and Validation

The AQI was calculated using the CPCB-standardized linear interpolation method. Comparison with ground truth data from the CPCB showed a Mean Absolute Error (MAE) of 2.87 AQI units across multiple stations. The classification accuracy for AQI categories (e.g., “Good,” “Moderate,” “Poor”) was over 95%.

TABLE I: Model Evaluation Metrics Across Five Air Pollutants

Pollutant	MAE (µg/m³)	RMSE (µg/m³)	R^2 Score (%)
PM2.5	4.8	6.2	90.1

PM10	7.1	9.5	89.7
NO_2	3.3	4.5	94.8
CO	0.18	0.26	90.3
O_3	5.4	6.1	92.5

C. Visualization and Temporal Trends

The system effectively visualized AQI trends using Seaborn and Matplotlib libraries. Weekly patterns, station-wise comparisons, and pollutant contributions were generated dynamically, as shown in Fig 2-.

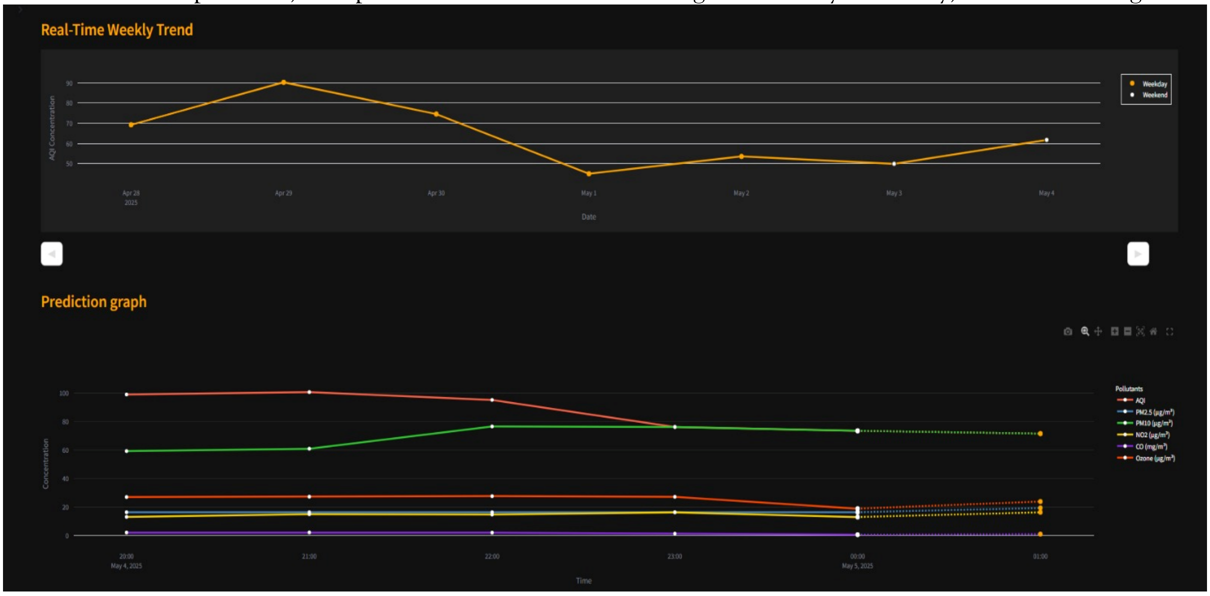

Fig. 3: Visualization of the forecasted pollutants

D. Chatbot Interaction Analysis

The Gemini 2.0-powered generative AI chatbot was evaluated for its performance in handling user queries, interpreting AQI values, and recommending personalized preventive actions.

- Intent recognition accuracy: 95
- Correctness of AQI explanation: 94
- Preventive recommendation relevance: 92
- Average response time: ¡2 seconds
- User satisfaction score: 4.7/5

ChromaDB memory allow the chatbot to maintain context across multi-turn interactions, significantly improving engagement and retention (Figure 3)

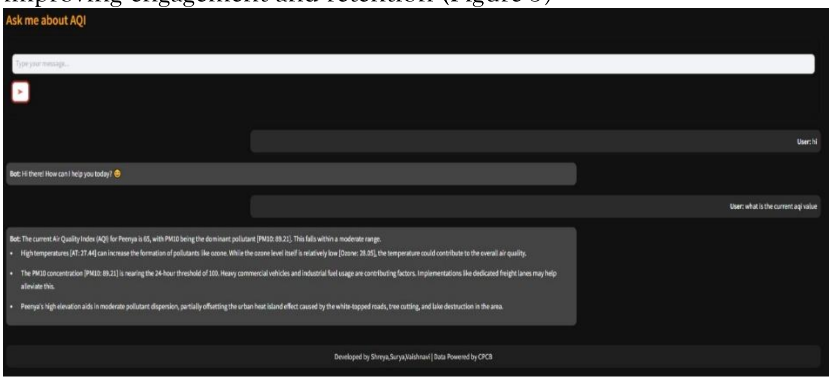

Fig. 4: Figure depicting chatbot implementation

E. Agent-based Simulations

Agent-based simulations enabled quantitative evaluation of environmental policies and, provided data-driven support for urban ban planning and pollution control.

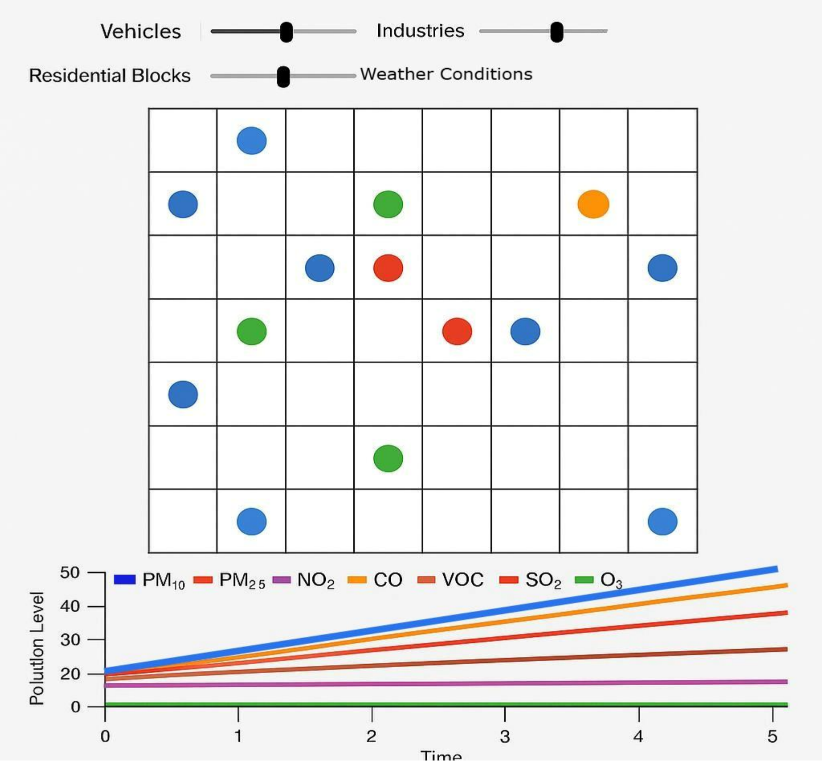

Fig. 5: Agent-based simulation result

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest: The authors declare no conflicts of interest.

Data availability: The data is web-scraped from pollutant and meteorological readings available on CPCB's CAAQMS stations using Selenium, and stored in a MongoDB database for real-time access. (https://airquality.cpcb.gov.in/ccr/#/caaqm-dashboard-all/caaqm-landing/data)

REFERENCES

[1] M. Dhilsath Fathima, Sashank Donavalli, and Harshitha Kambham, "Air Quality Prediction using Deep Learning models," in *Proc. 2024 Int. Conf. on Advancements in Power, Communication and Intelligent Systems (APCI)*, IEEE, 2024.

[2] R. F. Rahmat et al., "Identification of Safe Air Quality for Activity Using Long Short-Term Memory," in *Proc. 2022 6th Int. Conf. on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, IEEE, 2022.

[3] S. Sur, R. Ghosal, and R. Mondal, "Air Pollution Hotspot Identification and Pollution Level Prediction in the City of Delhi," in *Proc. 2020 IEEE 1st Int. Conf. for Convergence in Engineering (ICCE)*, IEEE, 2020.

[4] K. Thaweephol and N. Wiwatwattana, "Long short-term memory deep neural network model for PM2.5 forecasting in the Bangkok urban area," in *Proc. 2019 17th Int. Conf. on ICT and Knowledge Engineering (ICT&KE)*, IEEE, 2019.

[5] S. Bhatia, S. Sachdeva, and P. Goswami, "Air pollution prediction and hotspot detection using machine learning," *J. Stat. Manag. Syst.*, vol. 25, no. 7, pp. 1553–1564, 2022.

[6] R. Gupta, K. Khandal, and M. Kandan, "Air Quality Prediction in Smart Cities Using Regression Techniques," in *Proc. 2024 2nd Int. Conf. on Networking and Communications (ICNWC)*, IEEE, 2024.

[7] V. Chaudhary et al., "Time series based LSTM model to predict air pollutant's concentration for prominent cities in India," *UDM*, Aug. 2018.

[8] K. K. R. Samal, K. S. Babu, and S. K. Das, "Spatial-temporal prediction of air quality by deep learning and kriging interpolation approach," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 10, no. 5, 2023.

[9] Y. Cao et al., "A hybrid air quality prediction model based on empirical mode decomposition," *Tsinghua Sci. Technol.*, vol. 29, no. 1, pp. 99–111, 2023.

[10] Z. Wei et al., "Air Pollution Forecast Model Based on LSTM," in *Proc. 2021 6th Int. Conf. on Computational Intelligence and Applications (ICCIA)*, IEEE, 2021.
[11] G. Pandey, R. Sharma, and A. Kukker, "Bi-Directional Long Short-Term Memory Network (BiDLSTM) Model Based Air Pollution Prediction," in *Proc. 2023 2nd Int. Conf. on Futuristic Technologies (INCOFT)*, IEEE, 2023.
[12] K. Samal, K. Babu, and S. Das, "Spatio-temporal prediction of air quality using distance based interpolation and deep learning techniques," *EAI Endorsed Trans. Smart Cities*, vol. 5, no. 14, 2021.
[13] R. Dua et al., "Real time attention based bidirectional long shortterm memory networks for air pollution forecasting," in *Proc. 2019 IEEE 5th Int. Conf. on Big Data Computing Service and Applications (BigDataService)*, IEEE, 2019.
[14] R. Faiz, T. Agarwal, and D. Rajeswari, "Forecasting Air Pollution of Urban Chennai Localities using Artificial Intelligence," in *Proc. 2024 3rd Int. Conf. on Applied Artificial Intelligence and Computing (ICAAIC)*, IEEE, 2024.
[15] R. K. Grace, S. Vishnu, and V. Saveetha, "Air Quality Index Prediction using Bi-LSTM and Spider Monkey Optimization," in *Proc. 2024 7th Int. Conf. on Devices, Circuits and Systems (ICDCS)*, IEEE, 2024.
[16] S. Shetage, P. Palkar, and A. Kamble, "Chatbot for FAQ using NLP with ML," unpublished.
[17] S. SugunaSri et al., "A Question Answering System Application Integrated with Chatbot Using NLP," *Indian J. Sci. Technol.*, vol. 17, no. 29, pp. 2972–2980, 2024.
[18] F. Khennouche et al., "Revolutionizing generative pre-traineds: Insights and challenges in deploying ChatGPT and generative chatbots for FAQs," *Expert Syst. Appl.*, vol. 246, 2024, Art. no. 123224.
[19] S. Kale, G. Khaire, and J. Patankar, "FAQ-Gen: An automated system to generate domain-specific FAQs to aid content comprehension," *arXiv preprint arXiv:2402.05812*, 2024.
[20] E. Louis, "Exploring User-Desired Interaction in Conversational Generative AI Chatbots," 2024.
[21] R. A. Berkani, M. Mohamed et al., "An Intelligent Edge-Deployable Indoor Air Quality Monitoring and Activity Recognition Approach," *arXiv preprint arXiv:2311.XXXXX*, 2023.
[22] N. Bandara, S. Hettiarachchi, and P. Athukorala, "Airspec: An IoTempowered air quality monitoring system integrated with a machine learning framework to detect and predict defined air quality parameters," *arXiv preprint arXiv:2111.14125*, 2021.
[23] C. Lin et al., "Detecting Elevated Air Pollution Levels by Monitoring Web Search Queries: Deep Learning-Based Time Series Forecasting," *arXiv preprint arXiv:2211.05267*, 2022.