www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

PREDICTIVE MODELING OF AIR POLLUTION LEVELS VIA ARTIFICIAL INTELLIGENCE AND IOT DATA

P. Vemulamma[1*], Anukonti Sai Chandu[2], Bodige Sangeetha[2], Lingala Venkata Raji Reddy[2], Alle Akhil[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Sciences and Engineering

[1,2]Vaagdevi College of Engineering (UGC - Autonomous), Bollikunta, Warangal, Telangana.

[*]Corresponding author: P. Vemulamma (vemulamma@vaagdevi.edu.in)

ABSTRACT

This work presents an interactive, web-based platform for forecasting air-quality indices (AQI) by integrating a pre-trained deep learning model with a lightweight Flask application and a relational database for user management. The system enables users to register and authenticate through secure forms, which interface with a MySQL backend to store credential records. Upon successful login, end users access an intuitive HTML form, built with Jinja2 templates, to submit four key pollutant measurements Sulfur Oxides Index (SOI), Nitrogen Oxides Index (NOI), Respirable Particulate Index (RPI), and Suspended Particulate Matter Index (SPMI). These inputs are cast to floating-point values and packaged into a NumPy array that matches the input layer of a TensorFlow/Keras regression network, which has been trained offline on historical air-quality datasets. Session state is managed in-memory for simplicity, with a logout endpoint that resets the current user context. The modular architecture cleanly separates web routing, data persistence, and machine-learning logic, facilitating future enhancements: password hashing via Werkzeug utilities, cookie-based session management, input validation, and dynamic model updates. Finally, this prototype demonstrates how lightweight web frameworks can be combined with deep-learning models to deliver real-time environmental risk assessments through a user-friendly interface.

Keywords: Pollution monitoring, AQI prediction, Predictive modelling, Flask application, Web-based intelligence.

1. INTRODUCTION

Due to permanent degradation, environmental monitoring is mandatory both for industrial companies and for society to keep life parameters in a normal range. Ecological monitoring collects and analyses data on various environmental characteristics like temperature, humidity, gas, and dust concentrations. This action is critical in analyzing and maintaining ideal environmental conditions, particularly those related to indoor air quality, industrial processes, agriculture, and public health. Sensors are utilized in these applications to detect and quantify all types of environmental change. Businesses and government entities can use IoT devices to monitor and measure certain environmental factors in various settings [1]. The Internet of Things (IoT) is a network of actual objects, electronic devices, embedded systems, and other 'things' that gather and share data via the Internet using sensors and software applications to allow for remote monitoring and control. 'Things' are commonplace items that, regardless of the communication method used (RFID, Wi-Fi, Bluetooth, etc.), may be read, recognized, located, and addressed by information-sensing devices and/or controlled online anytime. The issue of air pollution is relevant and significant since it affects the entire natural ecosystem, and endangers people's health, being also associated with infectious disease transmission [2]. According to the World Health Organization, polluted air causes millions of deaths each year, primarily due to diseases such as heart

disease, stroke, chronic obstructive renal diseases, pulmonary disease, lung cancer, and acute respiratory infections [3]. To mitigate these consequences, it is critical to accurately assess air quality and anticipate its degradation in the short term caused by changes in wind directions and intensification, or other calamities. Platforms equipped with particle detectors may alert citizens to the rising quantities of pollen or dust. In such cases, it may be recommended that people avoid regions that may be hazardous to their health, take a different route, and find the nearest pharmacies where antihistamine medications may be purchased [4].

2. LITERATURE SURVEY

In recent years, significant progress has been achieved in the ability to monitor and, in some situations, anticipate air quality. Government monitoring stations, on the other hand, are accurate but have a limited number of locations and high operational costs [5]. To address these restrictions, researchers have combined IoT devices to improve flexibility and cost-effectiveness in monitoring air quality [6]. The Internet of Things (IoT) and machine learning have significantly improved air quality monitoring and prediction systems. IoT-enabled systems collect real-time data on air pollutants, which are then evaluated using machine learning techniques to forecast air quality levels [7,8]. The rapid expansion of IoT technology has revolutionized industries with remote monitoring and sophisticated analytics [9]. Monitoring air quality is crucial for resolving health issues and mitigating the effects of poor air quality on public health [10]. The rapidly expanding field of IoT monitoring indoor parameters includes sensor technology, data administration, user experience, health consequences, calibration, validation, and integration [11]. For example, Kumar Sai et al. [12] proposed an inexpensive IoT-based air quality monitoring system based on Arduino and the MQ series (specifically MQ135 and MQ7). These sensors detect ammonia, carbon dioxide, alcohol, smoke, and carbon monoxide. This arrangement allows for the cost-effective and adaptable display of contaminants in the air. The system analyzes air quality using data obtained from several sensors, proving the feasibility of using low-cost sensors for environmental monitoring. This strategy not only improves the availability of air quality monitoring, but also ensures that IoT can be used to address environmental challenges. The authors emphasize the importance of accessible air quality monitoring to raise awareness and improve public health.

Karnati [13] assessed IoT-based air pollution monitoring systems focusing on big data and machine learning. They highlighted the need for smart devices and advanced analytics for effective air quality control plans.

3. PROPOSED METHODOLOGY

The project is a web-based air-quality forecasting system built with Flask and a pre-trained Keras model. It allows users to register, log in, submit four key pollutant indices (SOI, NOI, RPI, SPMI), and receive both a numeric Air Quality Index (AQI) prediction and a corresponding category (e.g., Good, Moderate, Poor, etc.).

Key Components

- **Web Framework (Flask):** Routes handle user flows—home, login, signup, prediction form, result display, and logout—rendering Jinja2 templates for each view.
- **User Management:** MySQL (via PyMySQL) stores user credentials; simple username/password checks gate access. Registration prevents duplicates and enforces password confirmation.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

- **Machine-Learning Backend:** A Keras model (prediction_model.h5) is loaded at startup. When a user submits pollutant values, the model outputs a continuous AQI score.
- **AQI Classification:** A helper function maps the numerical output into human-readable categories (Good up to Hazardous), enabling clearer interpretation for end users.

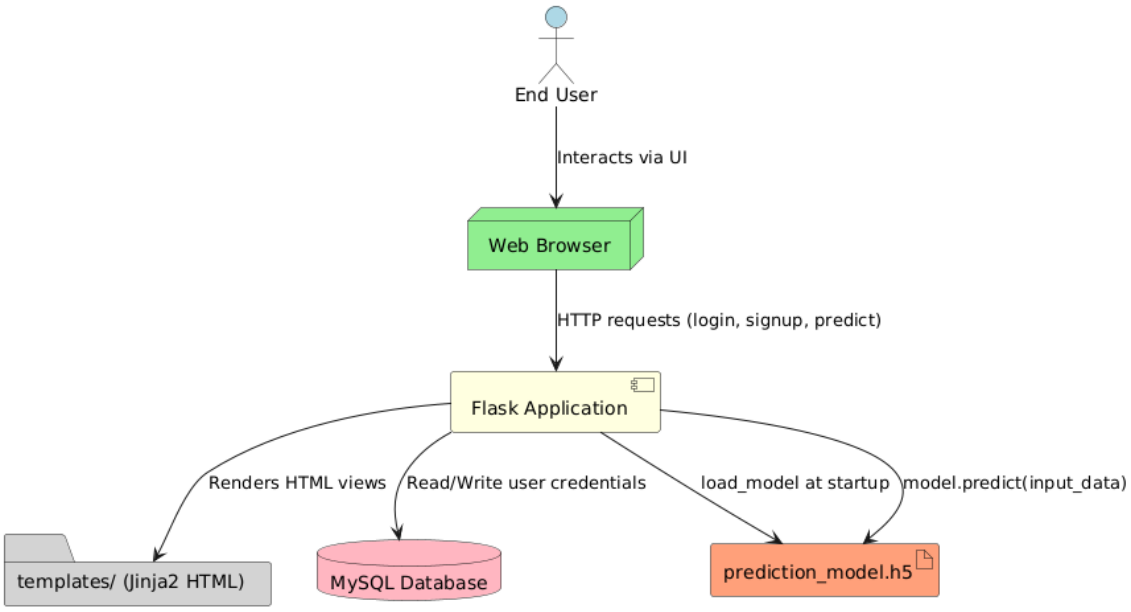

Fig. 1: System architecture of proposed AQI prediction model.

Proposed Workflow

The end-to-end workflow of the air-quality forecasting application can be broken down into four main stages:

1. **User Authentication & Session Management**
   - A visitor arrives at the root URL (/). If they're not logged in (the in-memory uname is None), links to **Login** and **Sign Up** are displayed.
   - On **Sign Up**, the user completes a registration form; the server verifies that passwords match and that the chosen username doesn't already exist in the MySQL register table before inserting the new record and redirecting to the login page.
   - On **Login**, submitted credentials are checked against the database. A successful match sets uname to the username, enabling personalized greetings ("Welcome, Mahesh") and granting access to the prediction form; a failed match reloads the login page with an error message.
2. **Input Collection & Validation**
   - Authenticated users click "Predict" (GET on /predict) to access a form with four fields—SOI, NOI, RPI, and SPMI.
   - When the form is submitted (POST to /predict), the server extracts each field via request.form, casts them to floats, and packages them into a single-row NumPy array. (At present there's no explicit range validation, but that could be added here.)

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

3. **Model Inference & Classification**
    - The Keras model (prediction_model.h5) is loaded into memory exactly once at application startup.
    - For each prediction request, the server calls model.predict(input_data) to compute a numeric Air Quality Index.
    - This raw output is passed to classify_air_quality(), which applies a series of threshold tests (≤50 → "Good", 51–100 → "Moderate", up to "Hazardous" for values >400) to produce a human-readable category.
4. **Result Rendering & Session Termination**
    - The server renders result.html, embedding both the numeric AQI and its textual classification.
    - The user can either run another prediction, navigate through the site, or click **Logout**, which clears the uname variable and returns them to the login screen.

This linear workflow—from authentication, through form-based input and ML inference, to categorized output—keeps concerns neatly separated (web routing vs. database vs. ML logic) and provides clear extension points (e.g., adding input validation, integrating hashed passwords, or enhancing the classification thresholds).

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

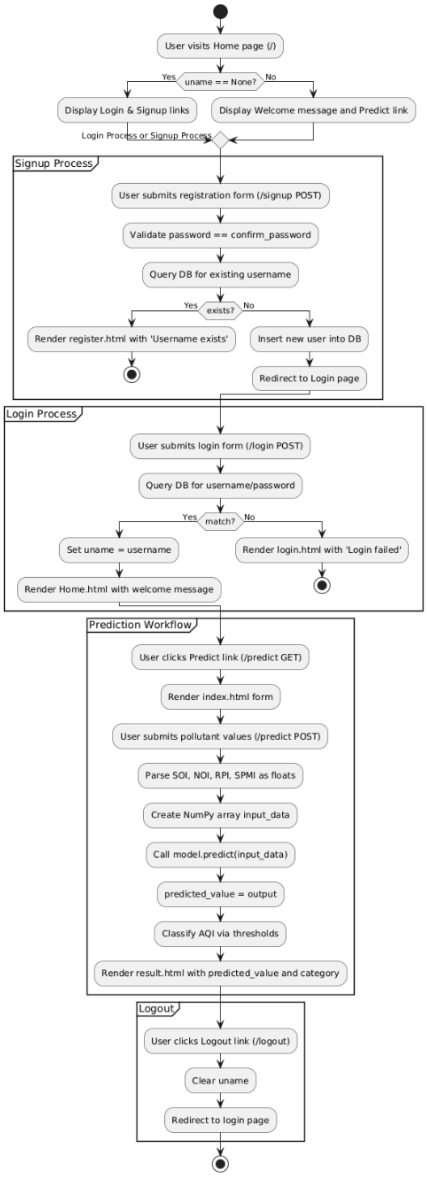

Fig. 2: Overall workflow of proposed AQI prediction.

4. RESULTS AND DISCUSSION

4.1 Implementation Description

The implementation phase of the Air Quality Prediction System involved the transformation of the detailed system design into an operational software product. The process followed a structured sequence of steps to ensure that the system performed all required tasks effectively, from data collection to deployment. Below is a detailed account of the major implementation steps:

Step 1: Dataset Collection and Preparation

The process began with the acquisition of a well-structured dataset consisting of air quality indicators such as SOI (Sulphur Oxide Index), NOI (Nitrogen Oxide Index), RPI (Respirable Particulate Index), and SPMI (Suspended Particulate Matter Index). These features were selected for their relevance to

Index in Cosmos
Apr 2025, Volume 15, ISSUE 2
UGC Approved Journal

determining the overall Air Quality Index (AQI). The dataset was cleaned by removing missing values and duplicate records. Data normalization was carried out to bring all features into a uniform scale, which enhanced the accuracy and stability of the predictive models.

Step 2: Model Training and Evaluation

A supervised machine learning approach was used for training. Multiple algorithms such as Linear Regression, Decision Tree, and Random Forest were tested. Each model was trained using the cleaned dataset, and its performance was evaluated using standard metrics such as R^2 score, MAE (Mean Absolute Error), and RMSE (Root Mean Squared Error). Based on evaluation results, the Decision Tree model was selected due to its interpretability and better performance on the given dataset. The trained model was serialized using Python's pickle library to allow integration with the web application.

Step 3: Backend Development with Flask

Flask, a lightweight Python web framework, was used for backend implementation. The backend handled routing, user authentication, input processing, and prediction logic. Routes were created to manage user login, signup, prediction submission, and result display. The Flask server was configured to load the serialized Decision Tree model and process incoming data from users in real-time, returning the AQI and its category classification as output.

Step 4: Frontend Interface Design

The frontend was developed using HTML, CSS, and JavaScript to provide a user-friendly interface. Static pages included home, login, signup, input form, and result display. Form elements were validated using client-side scripting to prevent empty or invalid entries. The design was made responsive to ensure compatibility across devices. Color-coded labels were used to represent AQI categories such as Good, Moderate, Poor, Unhealthy, Very Unhealthy, and Hazardous.

Step 5: Integration of Model with Web Application

The trained model was integrated into the backend logic. Once a user submitted input values through the form, the Flask server accepted the values, structured them into a suitable format, passed them to the model, and retrieved the predicted AQI value. This result was then mapped to an AQI category and rendered on the output page.

Step 6: User Session and Security Management

A basic user authentication system was implemented to allow users to register and log in. Passwords were encrypted using hashing algorithms to ensure security. User sessions were maintained using Flask's session management functionality. Input sanitization techniques were used to prevent injection attacks or unauthorized access.

Step 7: Testing and Validation

The application underwent thorough unit testing and functional testing. Model outputs were verified using test inputs to ensure accuracy. The user interface was tested across various browsers and devices. Backend routes were validated to ensure all inputs triggered the correct logic. Edge cases such as empty fields and invalid inputs were handled gracefully with proper messages.

Step 8: Deployment

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

The final version of the project was deployed on a local server for demonstration. All components were organized into appropriate directories—templates for HTML files, static for CSS/JS, and model for the serialized model file. The Flask application was run with proper configurations and all dependencies were managed using a requirements file.

4.2 Dataset Description

Each row in the dataset represents a specific instance of air quality measurements taken at a particular location and time. Here's a breakdown of the data columns used in this project:

- **stn_code**: Station code or identifier for the monitoring station.
- **sampling_date**: Date on which the air quality measurements were taken.
- **state**: State in which the monitoring station is located.
- **location**: Specific location where the air quality measurements were conducted.
- **agency**: Agency responsible for conducting the air quality monitoring.
- **type**: Type of air quality measurement or monitoring.
- **so2**: Sulphur dioxide (SO2) concentration in the air.
- **no2**: Nitrogen dioxide (NO2) concentration in the air.
- **rspm**: Respirable suspended particulate matter (RSPM) concentration in the air.
- **spm**: Suspended particulate matter (SPM) concentration in the air.
- **location_monitoring_station**: Name of the monitoring station's location.
- **pm2_5**: Fine particulate matter (PM2.5) concentration in the air.
- **date**: Date of the air quality measurement.

4.3 Results Description

The Home Page serves as the central access point of the Air Quality Prediction System, offering users a clean and intuitive interface. It provides navigation options such as Login, Signup, and Prediction access, allowing both new and registered users to proceed smoothly. The layout is user-friendly, with a welcoming message and a brief description of the system's purpose. Visually, it maintains a professional aesthetic with a responsive design suitable for all devices. The background image and color scheme are chosen to reflect environmental themes. The page dynamically updates the interface based on the user's login status, displaying a personalized greeting when logged in. It ensures seamless navigation to other modules of the system.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

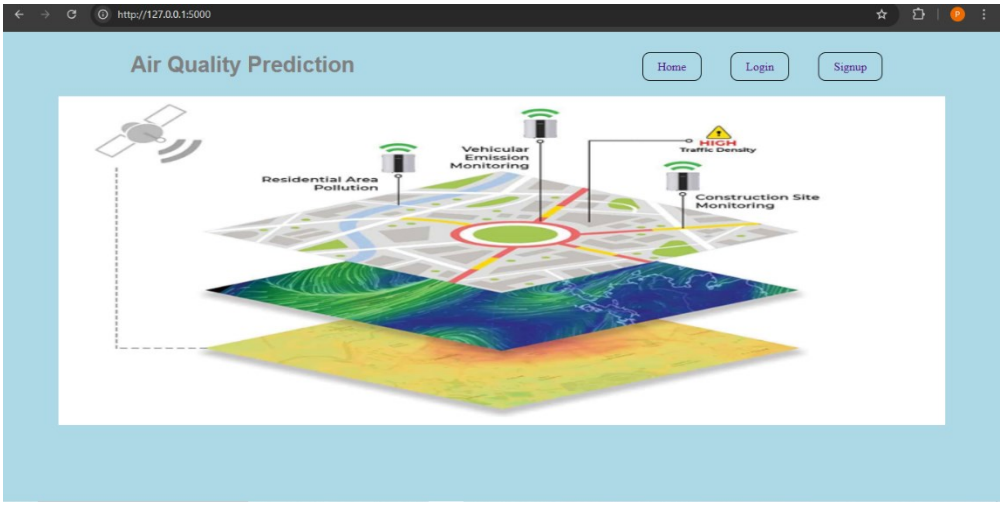

Fig. 3: Home page of proposed Flask application for AQI prediction.

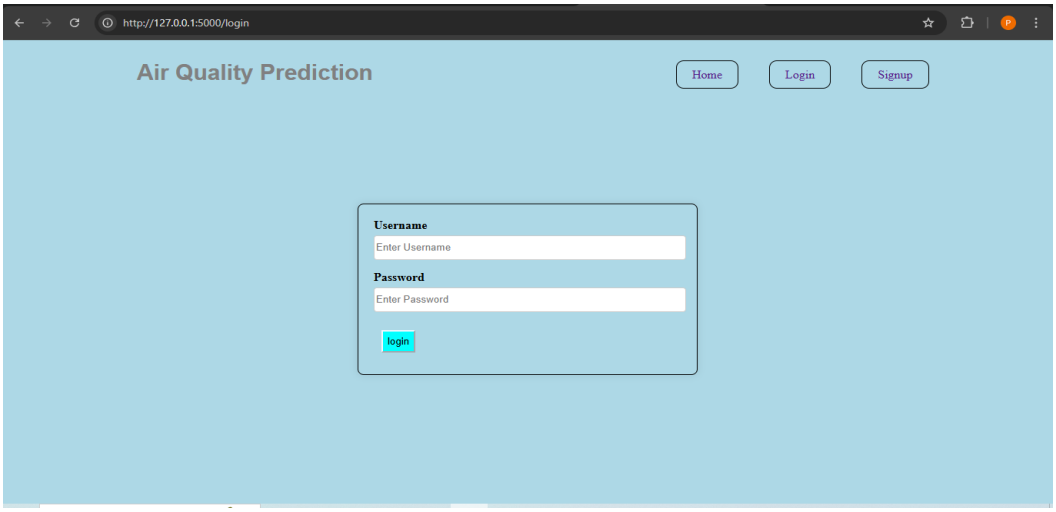

Fig. 4: Login page.

The Login Page functions as the secure entry portal for registered users of the Air Quality Prediction System. It includes input fields for entering the username and password, accompanied by a login button that initiates authentication. The page ensures user validation by checking credentials against the backend MySQL database. In case of incorrect details, it displays a clear error message prompting re-entry. The interface is simple and clean, prioritizing ease of use and clarity. It includes a link redirecting users to the Signup page if they do not already have an account. Upon successful login, users are directed to the Home Page with personalized access. This page ensures authorized access and protects user-specific functionalities within the system.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

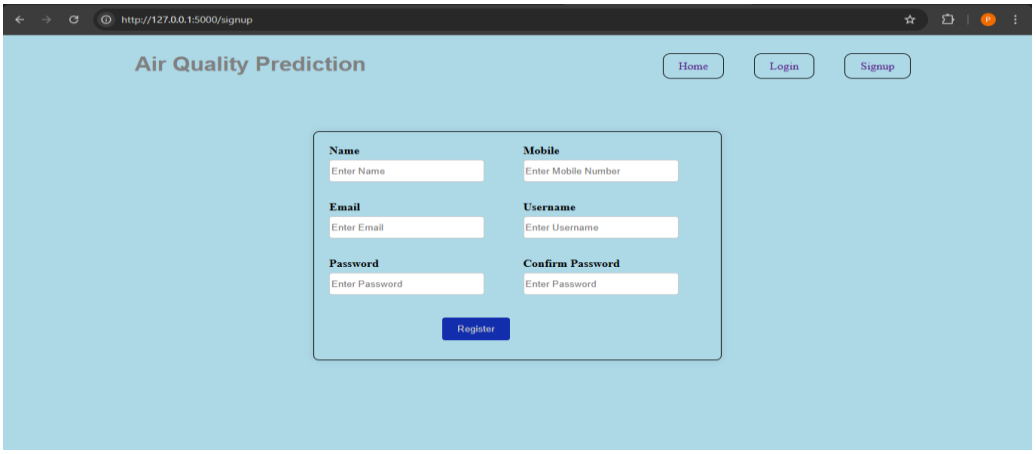

Fig. 5: Signup page.

The Signup Page allows new users to register for access to the Air Quality Prediction System by providing their personal and login details. It contains input fields for full name, mobile number, email, username, password, and confirmation of the password. The system performs validations to ensure data integrity, such as matching passwords and checking for existing usernames in the database. Upon successful registration, the user is redirected to the Login Page for authentication. The design emphasizes clarity and accessibility, ensuring users can complete the process without confusion. Error messages are displayed if any field is left incomplete or data is invalid. This page facilitates secure onboarding of new users and maintains accurate user records in the backend database.

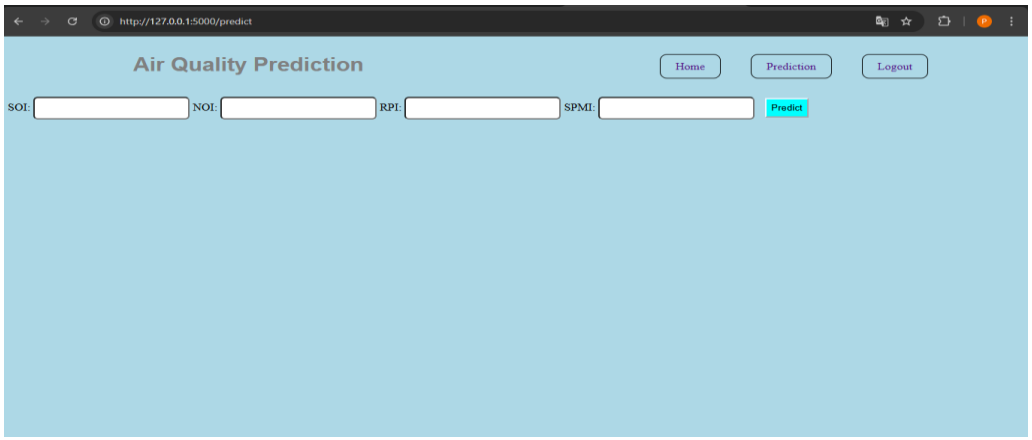

Fig. 6: Presents the web page development using Flask for AQI Prediction.

Figure 6 Shows an user interface for predicting air quality based on four input parameters: SOI (Sulfur Oxide Index), NOI (Nitrogen Oxide Index), RPI (Respirable Particulate Index), and SPMI (Suspended Particulate Matter Index). The form is styled using CSS to ensure proper alignment and spacing of input fields. A background image is set to enhance the visual appeal of the form.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

	so2	SOi
0	4.8	6.000
1	3.1	3.875
2	6.2	7.750
3	6.3	7.875
4	4.7	5.875

	no2	Noi
0	17.4	21.750
1	7.0	8.750
2	28.5	35.625
3	14.7	18.375
4	7.5	9.375

Fig. 7: Header of individual pollutant index for SO2 and NO2.

Figure 7 is a visualization for the classification of air quality based on the calculated AQI values. The classification likely involves different categories such as "good," "moderate," "poor," "unhealthy," "very unhealthy," and "hazardous." These categories indicate the level of pollution and associated health risks.

	state	SOi	Noi	Rpi	SPMi	AQI
0	Andhra Pradesh	6.000	21.750	0.0	0.0	21.750
1	Andhra Pradesh	3.875	8.750	0.0	0.0	8.750
2	Andhra Pradesh	7.750	35.625	0.0	0.0	35.625
3	Andhra Pradesh	7.875	18.375	0.0	0.0	18.375
4	Andhra Pradesh	5.875	9.375	0.0	0.0	9.375

Fig. 8: Header of AQI calculated from every data value.

	state	location	type	so2	no2	rspm	spm	pm2_5	SOi	Noi	Rpi	SPMi	AQI	AQI_Range
0	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	4.8	17.4	0.0	0.0	0.0	6.000	21.750	0.0	0.0	21.750	Good
1	Andhra Pradesh	Hyderabad	Industrial Area	3.1	7.0	0.0	0.0	0.0	3.875	8.750	0.0	0.0	8.750	Good
2	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.2	28.5	0.0	0.0	0.0	7.750	35.625	0.0	0.0	35.625	Good
3	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.3	14.7	0.0	0.0	0.0	7.875	18.375	0.0	0.0	18.375	Good
4	Andhra Pradesh	Hyderabad	Industrial Area	4.7	7.5	0.0	0.0	0.0	5.875	9.375	0.0	0.0	9.375	Good

```
Good              219643
Poor               93272
Moderate           56571
Unhealthy          31733
Hazardous          18700
Very unhealthy     15823
```

Fig. 9: Obtained classification of air quality as good, moderate, poor, unhealthy, very unhealthy, and Hazardous.

5.4 Comparative Analysis

Table 1 provides a comparison of two different machine learning models used for air quality prediction based on two evaluation metrics: Root Mean Squared Error (RMSE) and R-squared (R^2) score.

Index in Cosmos
Apr 2025, Volume 15, ISSUE 2
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Table 1: Comparison of ML models.

Model name	RMSE	R^2-score
LR	13.67	0.9847
Random Forest Regressor	1.16	0.999

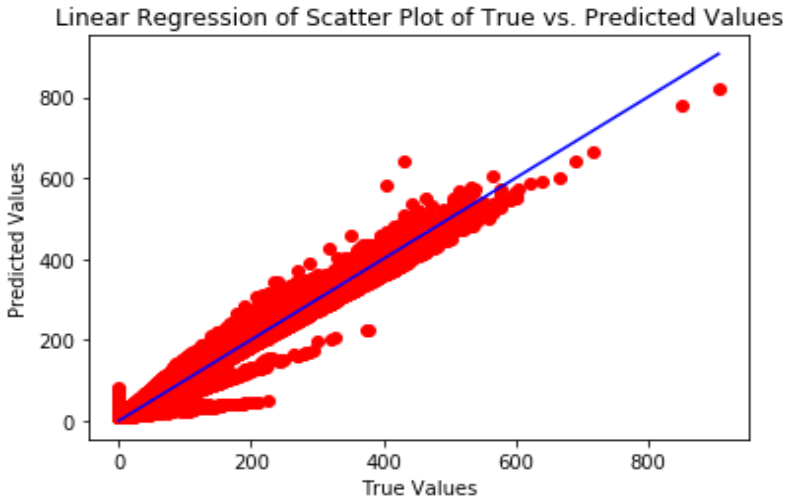

Fig. 10: Scatter plot of true and predicted values obtained using LR model.

Figure 10 is a scatter plot visualizes the performance of a LR model. In this plot, each point represents a data instance. The x-axis represents the true values (actual observations) of the target variable, while the y-axis represents the predicted values of the target variable made by the LR model. Each point on the plot corresponds to a data instance, where its position relative to the diagonal line (which represents a perfect prediction) indicates how well the model's predictions align with the actual data. If the points are close to the diagonal line, it suggests that the model's predictions are accurate.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

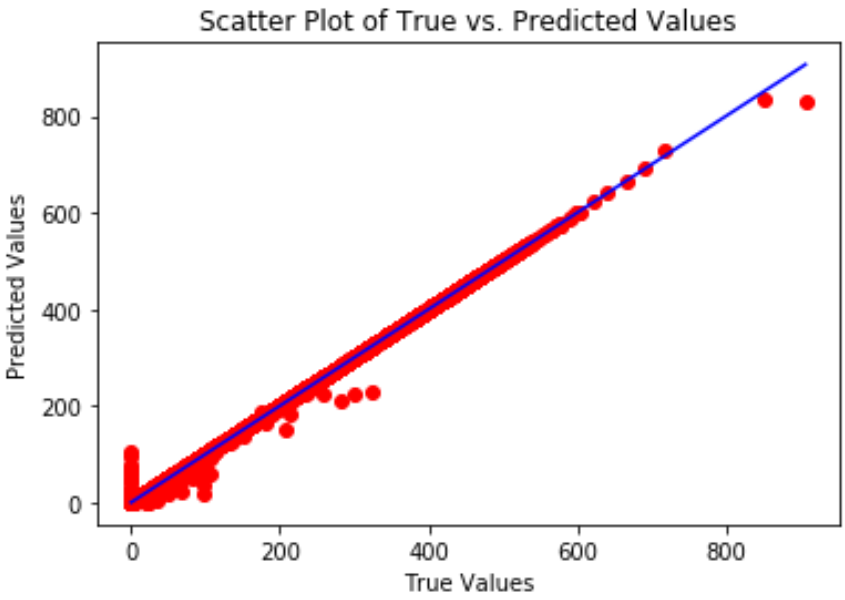

Fig. 11: Scatter plot of true and predicted values obtained using Random Forest regressor model.

In Figure 11, the scatter plot illustrates the performance of a Random Forest regressor model. Each point on the plot represents a data instance, where the x-axis shows the true values of the target variable, and the y-axis shows the predicted values made by the Random Forest model. The positioning of points relative to the diagonal line helps assess the accuracy of the model's predictions. From Figure 8, it indicates that the points cluster around the diagonal line are closely matches the actual values.

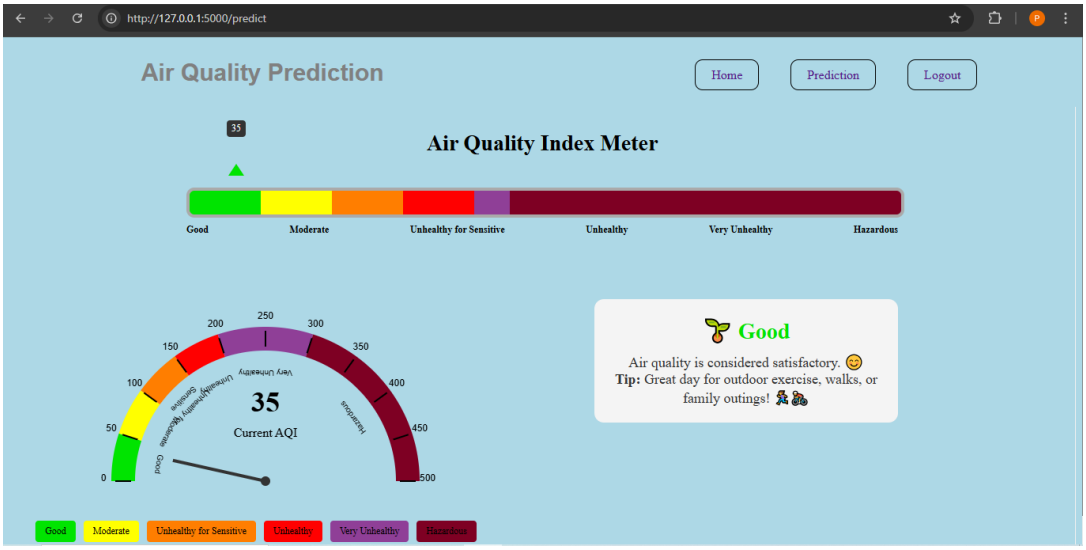

Fig. 12: Prediction output using Random Forest Regression model.

Figure 12 illustrates the output generated by the system after processing the user input through the Random Forest Regression model. The user enters values for specific air quality indicators such as SOI (Sulphur Oxide Index), NOI (Nitrogen Oxide Index), RPI (Respirable Particulate Index), and SPMI (Suspended Particulate Matter Index). Once submitted, the model evaluates the input and predicts an overall air quality value. Based on this numerical prediction, the system classifies the air quality condition into a predefined category like Good, Moderate, Poor, or Hazardous. The output is displayed

clearly on the screen with both the predicted value and its corresponding classification. This functionality enables users to easily understand the environmental condition in a given area. The prediction is presented with a user-friendly interface, allowing non-technical users to interpret the results effectively.

5. CONCLUSION

The project successfully demonstrates the feasibility of deploying a lightweight, user-friendly web application for real-time air-quality forecasting by combining Flask with a pre-trained Keras regression model. By encapsulating pollutant measurements into a concise NumPy input vector and executing an in-memory model inference, the system delivers low-latency AQI predictions alongside standardized categorical labels that facilitate end-user comprehension. The clear separation between web routing, template rendering, database operations, and ML inference not only streamlines development but also eases future enhancements such as introducing hashed password storage, robust session management, input validation, or automated model retraining pipelines. While the current implementation relies on plaintext credential handling and an in-memory session flag, it lays a strong architectural foundation for production hardening. Moreover, the threshold-based classification function can be readily adjusted to conform to evolving environmental guidelines or extended with probabilistic confidence intervals. Overall, this prototype highlights how modern deep-learning models can be wrapped in minimal web frameworks to empower stakeholders—ranging from environmental agencies to concerned citizens—with actionable air-quality insights via an accessible browser interface.

REFERENCES

[1] Fahmi, N.; Prayitno, E.; Musri, T.; Supria, S.; Ananda, F. An Implementation Environmental Monitoring Real-time IoT Technology. In Proceedings of the 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20 July 2022; pp. 1–4.

[2] Li, H.; Xu, X.L.; Dai, D.W.; Huang, Z.Y.; Ma, Z.; Guan, Y.J. Air pollution and temperature are associated with increased COVID-19 incidence: A time series study. Int. J. Infect. Dis. 2020, 97, 278–282.

[3] HEPA (Health and Energy Platform of Action). Call to Action to Increase Climate Resilience of Health Care Facilities & Air Quality Through Sustainable Energy. 2022. Available online: https://www.who.int/publications/m/item/call-to-action-to-increase-climate-resilience-of-health-care-facilities---air-quality-through-sustainable-energy (accessed on 10 April 2025).

[4] Florea, A.; Berntzen, L.; Johannessen, M.R.; Stoica, D.; Naicu, I.S.; Cazan, V. Low cost mobile embedded system for air quality monitoring. In Proceedings of the Sixth International Conference on Smart Cities, Systems, Devices and Technologies (SMART), Venice, Italy, 25–29 June 2017; pp. 25–29.

[5] Mokrani, H.; Lounas, R.; Bennai, M.T.; Salhi, D.E.; Djerbi, R. Air Quality Monitoring Using IoT: A Survey. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 127–134.

[6] Tran, Q.; Dang, Q.; Le, T.; Nguyen, H.-T.; Tan, L. Air Quality Monitoring and Forecasting System using IoT and Machine Learning Techniques. In Proceedings of the 2022 6th International Conference on Green Technology and Sustainable Development (GTSD 2022), Nha Trang City, Vietnam, 29 July 2022; pp. 786–792.

[7] Lalitha, K.V.; Naveen, A.V.; Supriya, A.S.V.; Nagalakshmi, P.S.R.; Kantham, P.S. Realtime Air Quality Evaluator Using Iot and Machine Learning. Int. J. Eng. Res. Technol. 2024, 13.

[8] Khanna, A.; Kaur, S. Internet of Things (IoT), Applications and Challenges: A Comprehensive Review. Wirel. Pers Commun. 2020, 114, 1687–1762.
[9] Saini, J.; Dutta, M.; Marques, G. Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review. Int. J. Environ. Res. Public Health 2020, 17, 4942.
[10] Abdulmalek, S.; Nasir, A.; Jabbar, W.A.; Almuhaya, M.A.M.; Bairagi, A.K.; Khan, M.A.; Kee, S.H. IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. Healthcare 2022, 10, 1993.
[11] Gangwar, A.; Singh, S.; Mishra, R.; Prakash, S. The State-of-the-Art in Air Pollution Monitoring and Forecasting Systems Using IOT, Big Data, and Machine Learning. Wirel. Pers. Commun. 2023, 130, 1699–1729.
[12] Sai, K.B.K.; Mukherjee, S.; Sultana, H.P. Low Cost IoT Based Air Quality Monitoring Setup Using Arduino and MQ Series Sensors with Dataset Analysis. Procedia Comput. Sci. 2019, 165, 322–327
[13] Karnati, H. IoT-Based Air Quality Monitoring System with Machine Learning for Accurate and Real-time Data Analysis. arXiv 2023, arXiv:2307.00580.