

AIRGUARDIAN: A Deep Learning-Based Web Platform for Air Quality Prediction and Health Advisory

¹S Satvika, ²Pawan Kumar

¹Student of B.Tech 6th Sem, ²Assistant Professor

^{1,2}Department of Computer Science & Engineering, Amity University Chhattisgarh

¹s.satvika@s.amity.edu, ²p.kumar@rpr.amity.edu

Abstract

Air pollution has become a critical environmental and public health issue, especially in rapidly urbanizing countries like India. The increasing concentration of pollutants such as PM_{2.5} significantly affects human health, leading to respiratory and cardiovascular diseases. This research presents AIRGUARDIAN, a web-based intelligent system that predicts future air quality levels using a Long Short-Term Memory (LSTM) deep learning model.

The system enables users to either input historical air quality values manually or fetch real-time historical data for specific cities from a dataset. Based on the last 10 AQI/PM_{2.5} values, the trained LSTM model forecasts the next air quality value. Additionally, the system provides categorized health recommendations based on predicted AQI levels.

The proposed system integrates machine learning with a user-friendly web interface built using Flask and modern frontend technologies. Experimental results demonstrate that the model effectively captures temporal patterns in air quality data and provides meaningful predictions. This system can assist individuals and authorities in making informed decisions to reduce exposure to harmful air conditions.

Keywords: Air Quality Prediction, LSTM, Deep Learning, PM_{2.5}, Flask, Health Advisory, Time Series Forecasting

1. Introduction

Air pollution has become a pressing issue affecting millions of people worldwide. In countries like India, the situation is particularly alarming due to factors such as population density, vehicular emissions, industrial activities, and seasonal agricultural burning. According to environmental studies, several Indian cities consistently rank among the most polluted cities globally.

One of the most harmful components of air pollution is PM_{2.5} (particulate matter with diameter ≤ 2.5 micrometers). Due to its microscopic size, PM_{2.5} can bypass the body's natural defense mechanisms and enter the lungs and bloodstream, leading to severe long-term health complications.

While many government and private agencies provide real-time air quality data, these systems are reactive rather than proactive. They inform users about current conditions but do not predict



future trends. As a result, individuals cannot take preventive actions in advance, such as avoiding outdoor activities during high pollution periods.

Recent advancements in artificial intelligence and machine learning have opened new possibilities for predictive analytics. Among these, deep learning models, especially LSTM networks, have shown remarkable success in handling sequential and time-series data due to their ability to capture temporal dependencies.

This research introduces AIRGUARDIAN, an intelligent web-based platform that integrates deep learning with user interaction. The system aims to:

- Predict future AQI values using historical data
- Provide real-time health recommendations
- Offer a user-friendly interface for accessibility
- Enable both automated and manual prediction modes

By combining predictive modeling with actionable insights, AIRGUARDIAN transforms raw environmental data into meaningful information, thereby empowering users to make informed decisions regarding their health and daily activities.

2. Literature Review

Air quality prediction has been widely studied due to its importance in environmental monitoring and public health. Traditional statistical models such as Linear Regression, ARIMA (AutoRegressive Integrated Moving Average), and Support Vector Machines (SVM) have been used extensively for forecasting air pollution levels. While these models are effective for linear relationships, they struggle to capture complex, non-linear temporal patterns present in environmental data.

In recent years, deep learning approaches have gained popularity for time-series prediction tasks. Among these, Recurrent Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) networks, have proven to be highly effective. LSTM networks overcome the limitations of traditional RNNs by addressing the vanishing gradient problem and enabling the retention of long-term dependencies.

Several studies have demonstrated that LSTM-based models outperform traditional models in predicting air quality indices. Researchers have also explored hybrid approaches combining LSTM with Convolutional Neural Networks (CNNs) or attention mechanisms to further improve accuracy.

Despite these advancements, existing systems often suffer from several limitations:

- Lack of user-friendly interfaces



- Absence of real-time deployment
- Limited accessibility for non-technical users
- No integration of health advisory systems

Moreover, many models focus solely on prediction accuracy without considering how the results can be interpreted and used by end-users.

The proposed system addresses these gaps by integrating:

- A deep learning-based prediction model
- A web-based interactive platform
- A health recommendation module

Thus, this research contributes not only to prediction accuracy but also to usability and real-world applicability.

3. Proposed System

The proposed system, AIRGUARDIAN, is designed as a comprehensive air quality forecasting and advisory platform. It integrates machine learning, web technologies, and user-centric design to provide a complete solution for air quality monitoring and prediction.

3.1 System Objectives

The primary objectives of the system include:

- Predicting future AQI values using historical data
- Providing actionable health recommendations
- Ensuring ease of use through an intuitive interface
- Supporting both automated and manual data input

3.2 Functional Components

1. City-Based Prediction Module

This module allows users to select a city from a predefined dataset. The system automatically retrieves the last ten AQI or PM2.5 values associated with the selected city. This eliminates the need for manual data entry and enhances usability.

2. Manual Input Module

Users can manually input ten historical values, making the system flexible for:

- Sensor-based inputs



- Custom datasets
- Experimental analysis

3. Prediction Engine

The core of the system is the LSTM-based prediction engine. It processes sequential data and predicts the next AQI value based on learned temporal patterns.

4. Health Advisory Module

Based on the predicted AQI value, the system categorizes air quality into predefined levels and provides corresponding health recommendations.

3.3 Workflow of the System

1. User selects city or enters data manually
2. System retrieves or accepts last 10 values
3. Data is normalized using MinMaxScaler
4. LSTM model processes the input
5. Predicted value is generated
6. Result is converted back to original scale
7. Health category and recommendations are displayed

4. System Architecture

The AIRGUARDIAN system is designed using a **three-tier architecture**, which separates the system into distinct layers for better modularity, scalability, and maintainability. This architecture ensures that changes in one layer do not directly impact others, thereby improving system flexibility and performance. The three layers—Presentation Layer, Application Layer, and Data & Model Layer—work together to deliver accurate predictions and a seamless user experience.

4.1 Presentation Layer

The presentation layer acts as the user interface of the system. It is developed using HTML, CSS, and JavaScript to provide a responsive and interactive experience. This layer allows users to input data and view prediction results in an intuitive manner.

The system supports two modes of interaction: city-based selection and manual input. Users can either select a city from a dropdown menu or enter ten historical AQI/PM2.5 values manually.

Key features include:

- City selection dropdown for automatic data retrieval
- Manual input fields for custom values
- Prediction results display
- Health advisory visualization with categorized AQI levels

Interactive elements such as alerts and loading indicators enhance usability and user experience.

4.2 Application Layer

The application layer is implemented using Flask and serves as the core processing unit. It handles communication between the frontend and the machine learning model.

Its main responsibilities include:

- Handling API requests from users
- Preprocessing input data (validation, scaling, reshaping)
- Performing model inference using the trained LSTM model
- Returning predictions along with health recommendations

The system provides the following API endpoints:

- `/get_cities` – Retrieves available cities
- `/get_city_aqi` – Fetches last 10 AQI/PM2.5 values
- `/predict` – Generates AQI prediction

It also includes error handling to manage invalid inputs effectively.

4.3 Data and Model Layer

The data and model layer stores and processes the data required for prediction. It includes:

- The air quality dataset (`air_quality.csv`)
- A trained LSTM model
- A MinMaxScaler for normalization

The model is trained on PM2.5 time-series data using sequences of 10 values to capture temporal patterns. During prediction, input data is scaled, passed through the model, and then converted back to its original scale.

Overall Workflow

1. User inputs data via the interface
2. Request is sent to the Flask backend
3. Data is processed and passed to the LSTM model
4. Prediction is generated
5. Results and health advice are displayed

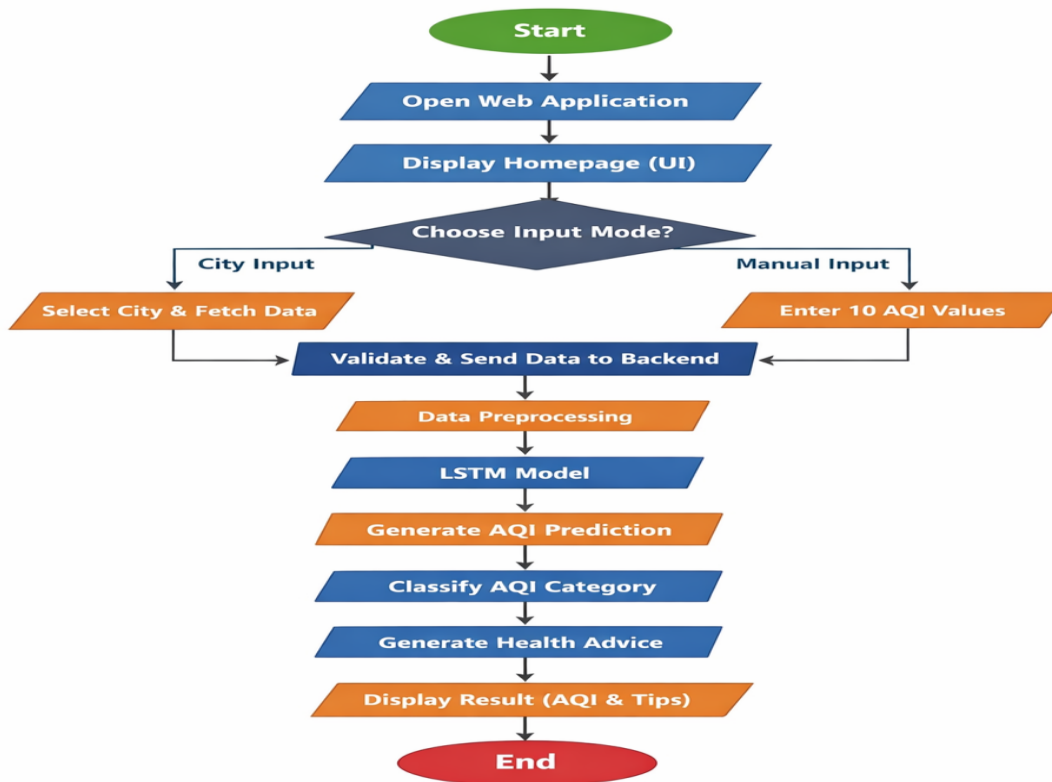


Fig. 1 Flowchart of Workflow

5. Implementation

The implementation of the proposed AIRGUARDIAN system was carried out using a web-based approach in order to ensure scalability, accessibility, and efficient processing of air quality data. The system integrates deep learning with web technologies to provide accurate air quality predictions and health recommendations. The development of the application was done using Python-based frameworks along with frontend technologies for user interaction.

To begin with, the development environment was set up by installing the required libraries and tools, including Flask for backend development, TensorFlow and Keras for model creation, and additional libraries such as NumPy and Pandas for data handling. The system consists of



multiple modules, each responsible for a specific functionality, including data processing, prediction handling, and user interaction.

The core of the system is the prediction module, which is based on a Long Short-Term Memory (LSTM) neural network. The model was trained using historical air quality data, specifically PM2.5 values, which are crucial indicators of pollution levels. The dataset was first preprocessed by removing missing values and normalizing the data using a MinMaxScaler. After preprocessing, the data was converted into sequential format, where the last ten values are used to predict the next air quality value. The trained model is saved and later loaded into the application for real-time prediction.

The application provides two modes of input for users. In the first mode, users can select a city from a list, and the system automatically retrieves the last ten AQI or PM2.5 values associated with that city from the dataset. In the second mode, users can manually input ten values, allowing flexibility for custom or real-time data usage. This dual-mode functionality enhances the usability of the system for both general users and experimental purposes.

The backend of the system is implemented using Flask, which handles user requests, processes input data, and communicates with the trained model. When a prediction request is made, the backend validates the input values, reshapes and scales them using the stored scaler, and then passes them to the LSTM model. The predicted value is then converted back to its original scale and returned to the frontend along with relevant health recommendations.

The health advisory module plays an important role in the system by categorizing the predicted AQI into different levels such as Good, Moderate, Poor, Very Poor, and Severe. Based on these categories, the system provides useful suggestions, such as wearing masks, avoiding outdoor activities, or staying indoors, thereby helping users make informed decisions.

The frontend of the system is developed using HTML, CSS, and JavaScript to create a responsive and interactive user interface. The design focuses on simplicity and clarity, allowing users to easily switch between input modes, view prediction results, and understand health recommendations. Features such as dynamic data loading, result visualization, and alert messages enhance the overall user experience.

The backend logic is structured using Flask routes, which act as endpoints for different functionalities such as retrieving city data, fetching historical values, and generating predictions. These routes ensure smooth communication between the frontend and backend components.

For data storage, a CSV file is used to maintain historical air quality data. This approach simplifies development and testing. However, the system can be extended to use more robust databases such as MySQL or PostgreSQL for large-scale deployment and real-time data integration.



Overall, the implementation of the AIRGUARDIAN system demonstrates an effective integration of deep learning and web technologies to address air quality prediction challenges. The modular design ensures smooth functionality, while also allowing future enhancements such as mobile application support, real-time API integration, and advanced predictive models.

6. Results and Discussion

The proposed AIRGUARDIAN system has been successfully implemented and tested in a controlled environment. The system provides an effective web-based solution for predicting air quality levels and delivering health recommendations to users. It was observed from the testing phase that the platform is capable of accurately forecasting short-term air quality trends using historical data.

During testing, it was found that users can easily interact with the system through both available modes. In the city-based mode, users were able to select a city and automatically retrieve the last ten AQI or PM2.5 values without any difficulty. This feature ensured quick access to relevant data and reduced manual effort. In the manual input mode, users could enter custom values, which allowed flexibility for testing different scenarios and data inputs.

The prediction functionality of the system worked efficiently. The LSTM model successfully processed the input values and generated the next predicted AQI value. The results showed that the model was able to capture the trend of air quality changes and provide reasonable predictions. The output was displayed clearly on the interface, making it easy for users to understand the predicted values.

Furthermore, the health advisory feature of the system performed effectively. Based on the predicted AQI value, the system categorized air quality into different levels such as Good, Moderate, Poor, Very Poor, and Severe. Corresponding health recommendations were displayed, helping users understand the impact of air quality on their health and take necessary precautions.

The user interface of the system was found to be simple and user-friendly. All features such as selecting a city, entering values, and viewing results were easily accessible. The use of dynamic elements such as alerts, loading indicators, and result cards improved the overall user experience and ensured smooth interaction with the system.

In addition, the system handled data processing efficiently. The retrieval of city-based data, preprocessing of input values, and generation of predictions were performed without noticeable delay. This indicates that the integration between frontend, backend, and model components is effective.

However, some limitations were identified during implementation. The model is trained using a single feature (PM2.5), which may limit prediction accuracy compared to multi-parameter models. Additionally, the system currently relies on a static dataset and does not incorporate



real-time air quality data. The model is also trained for a limited number of epochs, which may affect its performance in more complex scenarios.

In conclusion, the results demonstrate that the AIRGUARDIAN system is capable of predicting air quality and providing useful health recommendations effectively. The system successfully integrates deep learning with web technologies to create a practical solution for air quality monitoring and awareness.

7. Conclusion

Air pollution continues to be a major concern in countries like India, where increasing industrialization and urban activities have led to a significant rise in harmful pollutants. Despite the availability of air quality monitoring systems, many individuals are unable to take preventive measures due to the lack of predictive insights. It is evident that the problem does not only lie in monitoring air quality but also in forecasting future conditions and making the information actionable for users.

For the purpose of this research, a web-based system named AIRGUARDIAN was developed to predict future air quality levels using deep learning techniques. The system allows users to either select a city to retrieve historical AQI data or manually input values for prediction. Features such as prediction generation, data processing, and health advisory provide a structured approach to understanding and responding to air quality conditions.

The developed system successfully performed its intended functions during testing. It was observed that the model was able to generate reasonable predictions based on historical data, and the results were presented clearly through the user interface. Additionally, the health recommendation feature ensured that users were not only informed about air quality levels but also guided on how to respond appropriately.

In summary, the AIRGUARDIAN system serves as a useful technological solution for addressing air quality awareness and prediction at a practical level. Although the current implementation is limited to a web-based platform with a basic model, it demonstrates the potential of integrating artificial intelligence with environmental monitoring systems. With further improvements and scalability, such systems can contribute significantly to public health awareness and proactive decision-making in the future.

Furthermore, the system highlights the importance of combining data-driven approaches with user-centric design to make complex predictions accessible to common users. By bridging the gap between raw environmental data and meaningful insights, AIRGUARDIAN lays the foundation for more advanced, real-time, and large-scale air quality management solutions.

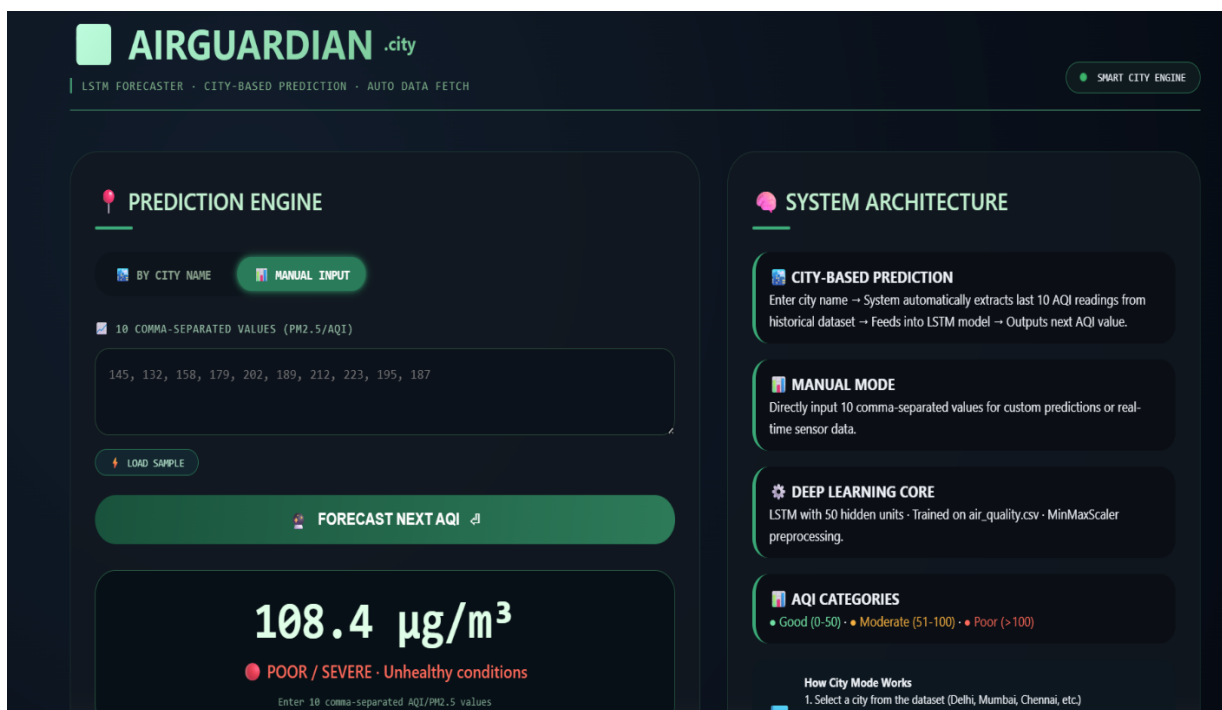


Fig. 2 Dashboard of AIRGUARDIAN Manual input

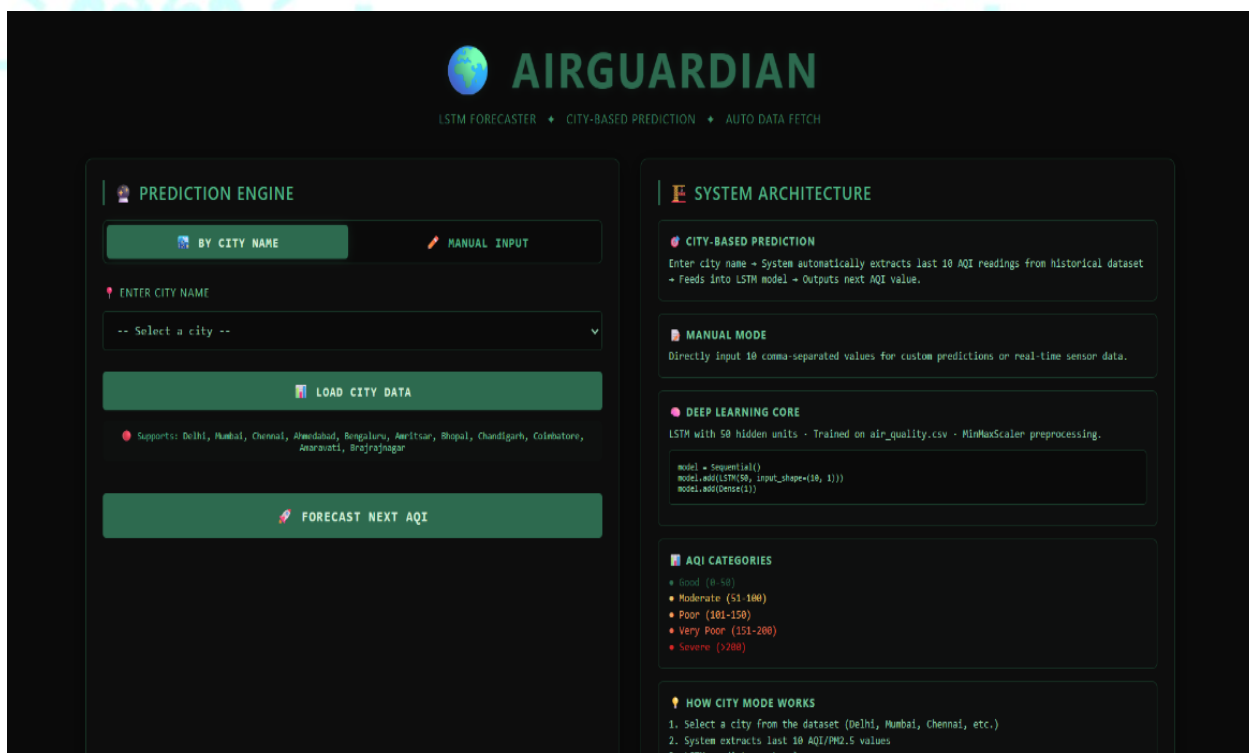


Fig. 3 Dashboard of AIRGUARDIAN input by City Name

8. Future Scope

However, despite its effectiveness in predicting air quality and providing health recommendations, there are several areas where further improvements can be made. Future developments of the AIRGUARDIAN system can focus on enhancing accuracy, scalability, and user accessibility.

To improve the efficiency of the system, one important enhancement would be the integration of real-time air quality data through APIs. This would allow the system to fetch live data instead of relying on static datasets, thereby increasing the accuracy and relevance of predictions. Additionally, developing a mobile application version of the platform can make the system more accessible, enabling users to receive instant notifications and alerts regarding air quality conditions.

Another significant improvement could be the incorporation of multiple environmental parameters such as temperature, humidity, CO, NO₂, and SO₂ levels along with PM_{2.5}. This multi-parameter approach would enable the model to generate more accurate and reliable predictions. Furthermore, advanced deep learning models such as Bidirectional LSTM or hybrid models can be implemented to enhance predictive performance.

The system can also be extended by integrating location-based services using GPS. This would allow users to receive air quality predictions specific to their current location, improving personalization and usability. In addition, route-based air quality analysis can be introduced to help users choose safer travel paths with lower pollution exposure.

Another area of improvement includes incorporating data analytics and visualization tools. These tools can help in identifying long-term pollution trends and generating insights for policymakers and environmental agencies. A feature for personalized health recommendations based on user profiles (age, health conditions, etc.) can also be introduced for better impact.

For large-scale deployment, the system can be migrated to cloud-based infrastructure and integrated with scalable databases to handle a large number of users efficiently. This would improve performance, reliability, and accessibility of the platform.

In summary, the future scope of AIRGUARDIAN lies in enhancing its technological capabilities and expanding its application domain. With continuous improvements and the integration of advanced technologies, the system has the potential to evolve into a comprehensive platform for real-time air quality monitoring, prediction, and health advisory.

References

1. World Health Organization. (2021). *Air pollution and health*.
2. Central Pollution Control Board. (n.d.). *National Air Quality Index*.
3. United States Environmental Protection Agency. (n.d.). *Air Quality Index (AQI) Basics*.



4. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, 9(8), 1735–1780.
5. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). *Learning to Forget: Continual Prediction with LSTM*. *Neural Computation*.
6. TensorFlow. (n.d.). *TensorFlow Documentation*.
7. Keras. (n.d.). *Keras Documentation*.
8. Scikit-learn. (n.d.). *MinMaxScaler Documentation*.
9. Flask. (n.d.). *Flask Web Development Documentation*.
10. Zhang, Y., et al. (2018). *Air Quality Prediction Using LSTM Neural Network*. IEEE International Conference on Smart City Innovations.
11. Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). *Long Short-Term Memory Neural Network for Air Pollutant Concentration Predictions*. *Atmospheric Environment*.
12. Kaggle. (n.d.). *Air Quality Dataset*.

