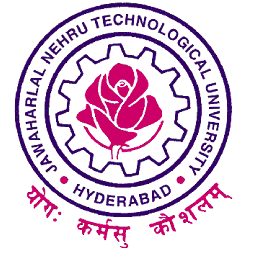
**AIR QUALITY PREDICTION USING MACHINE LEARNING**

**A Real-Time Research Project Report**

***Submitted to***

******

# Jawaharlal Nehru Technological University

Hyderabad

*In partial fulfillment of the requirements for the*

*award of the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

By

**Sallagarige Spoorthi (23VE1A66J1)**

**Kataram Navya Sree (23VE1A66G2)**

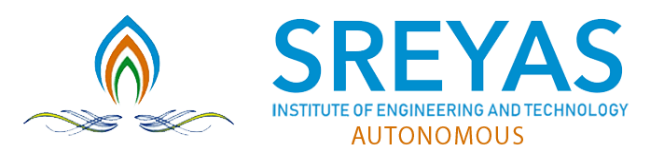
**Ratan Manisha (23VE1A66H8)**

**NTejavath ChandraShekar (23VE1A66J7)**

**Under the Guidance of**

**Mrs.B.Spandana**

**Assistant Professor**



**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NAAC “A” Grade & NBA| Hyderabad | PIN: 500068

**Certificate**

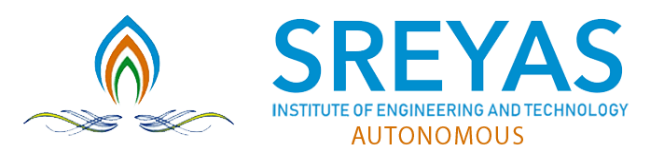
This is to certify that the Real-Time Research Project Report on***“AIR QUALITY PREDICTION USING MACHINE LEARNING***”submitted by **S.Spoorthi, K.NavyaSree, R.Manisha,T.ChandraShekar** bearing Hall Ticket No’s.**23VE1A66J1, 23VE1A66G2, 23VE1A66H8, 23VE1A66J7** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Artificial Intelligence & Machine Learning** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad for the academic year 2023-24 is a record of bonafide work carried out by him / her under our guidance and Supervision.

**Internal Guide Head of the Department**

**Mrs.B.Spandhana Dr. A. Swathi**

**Asst. Professor**

**Project Coordinator**



**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

Approved by AICTE, New Delhi | Affiliated to JNTUH, Hyderabad | Accredited by NAAC “A” Grade & NBA| Hyderabad | PIN: 500068

**DECLARATION**

We **S.Spoorthi, K.NavyaSree,R.Manisha,T.ChandraShekar**, bearing Roll No’s **23VE1A66J1, 23VE1A66G2, 23VE1A66H8, 23VE1A66J7** hereby declare that the Project titled "***AIR QUALITY PREDICTION USING MACHINE LEARNING***” done by us under the guidance of **Mrs.B.Spandana**, which is submitted in the partial fulfillment of the requirement for the award of the B.Tech degree in **Artificial Intelligence & Machine Learning** at **Sreyas Institute of Engineering & Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

**S.Spoorthi 23VE1A66J1**

**K.NavyaSree 23VE1A66G2**

**R.Manisha 23VE1A66H8**

**T.ChandraShekar 23VE1A66J7**

**ACKNOWLEDGEMENT**

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success.

We take this opportunity to acknowledge with thanks and a deep sense of gratitude to **Mr/Mrs. S.Spandhana Associate Professor**, for her constant encouragement and valuable guidance during the project work.

A Special vote of Thanks to **Dr. A. SWATHI (Head of the Department, AIML) and Mrs.B.Spandana (Project Co-Ordinator)** has been a source of Continuous motivation and support. They had taken time and effort to guide and correct me all through the span of this work.

We owe everything to the **Department Faculty, Principal** and the **Management** who made my term at Sreyas Institute of Engineering and Technology a stepping stone for my career. I treasure every moment I have spent in college.

Last but not the least, my heartiest gratitude to my parents and friends for their continuous encouragement and blessings. Without their support, this work would not have been possible.

**S.Spoorthi 23VE1A66J1**

**K.NavyaSree 23VE1A66G2**

**R.Manisha 23VE1A66H8**

**T.ChandraShekar 23VE1A66J7**

# CHAPTERS INDEX

# CHAPTER 1. INTRODUCTION…………………………………………….8

## Abstract………………………………………………………….…………8

1.2 Problem statement…………………………………………………………..8

## **CHAPTER 2. LITERATURE SURVEY……………………..………..……..9-12**

## 2.1Existing System……...………….....................................................................9-10

## 2.2 Proposed System….........................................................................................10-12

## **CHAPTER 3. SYSTEM DESIGN….…………………………………………13-17**

## 3.1 Importance of Design……………………………………………………….13

## 3.2 System Architecture…………………………………………………………14-15

## 3.3Functional Requirements…………………………………………………….15-17

## **CHAPTER 4. IMPLEMENTATION…………………………………………18-27**

4.1 Module Description………………………………………………………….18-21

4.2 Module Components…………………………………………………………21-27

4.2.1 Dataset (If any)……………………………………………………..21-22

4.2.2 Data Analysis and Visualization…………………………………...22-23

4.2.3 Web Interface………………………………………………………23

4.3 Sample Code…………………………………………………………23-27

**CHAPTER 5. RESULTS……………………………………………………….28-34**

**CHAPTER 6. CONCLUSION AND FUTURE SCOPE………………………34-38**

**CHAPTER 7. REFERENCES………………………………………………….38-40**

**PUBLICATION CERTIFICATE………………………………………………41**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig.No** | **Name of Figures** | **Page No** |
| **Fig.1** | **System Architecture** | **14** |
| **Fig.2** | **Distribution of Air Quality Index (AQI)** | **28** |
| **Fig.3** | **Top 10 Cities with the highest average AQI** | **29** |
| **Fig.4** | **Comparing the distributions of PM2.5 and PM10** | **30** |
| **Fig.5** | **Distribution of Air Quality Index (AQI) Buckets** | **31** |
| **Fig.6** | **Correlation Heat Map of PM2.5,PM10, AQI** | **32** |
| **Fig.7** | **AQI Trend Over Time as per date and Average AQI** | **33** |
| **Fig.8** | **AQI Forecating using ARIMA** | **33** |
| **Fig.9** | **AQI Forecating using LSTM** | **34** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No** | **Name of Table** | **Page No** |
| Table. 1 | Dataset explanation table | **21-22** |
| Table. 2 | Dataset explanation table | **22** |
| Table. 3 | Dataset explanation table | **22** |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Abstract**

Predicting air quality is a serious issue for the management of environmental health risks and the making of pollution control decisions. This abstract is concerned with a critical review of the recent developments in the use of machine learning techniques in air quality prediction, highlighting their efficacy, challenges, and future prospects. We all know that pollution in our country is increasing day by day due to which the death toll rate around the world is vigorously rising. But among these, one cause that is affecting us in a vigorous manner is "air pollution".

Air pollution is now becoming a major issue to human life and other living organisms. The air quality of our country is decreasing or can say is being affected day by day; itself this is a matter of concern to the health department. Due to the increase in the pollution level, the air quality index is increasing day by day. Air quality index of India is a highly statistical degree factor, that has been used in getting analysis for pollutant present in the atmosphere like NO2, Respirable Suspended Particulate Matter, SO2, Suspended particulate matter, etc. levels over a period of time.

Our idea basically is to perform better analysis of the air quality index by applying various algorithms of machine learning. Impact will be seen in the areas of urban planning and policy development, where the high accuracy predictions in air quality will help underline the designing of sustainable cities and setting up effective control measures against pollution. The project helps in creating healthier and more livable urban environments through its data-driven insights. The air quality prediction project is one such effort at using the power of machine learning to help alleviate some of the problems arising from air pollution and sustain better health outcomes for urban populations.

* 1. **Problem Statement**

Air pollution is a growing concern in many urban and semi-urban areas. Timely

and accurate forecasting of air quality metrics like AQI is crucial for public health advisories

and policy formulation. This project aims to:

1. **Clean and pre-process** a Dataset containing daily air quality readings from various Indian cities.
2. **Analyze** key pollutants (e.g., PM2.5, PM10, NO2) and their impact on AQI using statistical and visualization techniques.
3. **Identify trends**, anomalies, and patterns in air quality data across cities and over time. 4.**Forecast future AQI values** for the next 30 days using:A classical time-series model: **ARIMA.**A deep learning-based approach: **LSTM (Long Short-Term Memory neural network)** using PyTorch
4. **Compare results** from both models and visualize the predicted AQI trends to understand their practical utility.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1Existing Systems**

A number of systems and models have been designed and developed for the prediction of pollutant levels and assessment of air quality in the area of air quality prediction. The working principle behind the majority of these systems is essentially based on statistical and machine learning techniques to analyze and forecast the air quality index. This section reviews the existing systems categorized into Time Series Analysis and Regression Models.

**2.1.1 Time Series Analysis**

Time series analysis is a statistical technique dealing with data points obtained at specific time intervals. In the case of air quality prediction, it becomes even more operational due to the fact that data associated with pollution is normally temporal in nature.

**ARIMA—AutoRegressive Integrated Moving Average**

**● Description:** ARIMA is one of the popular models for analyzing and predicting time series data. Combines autoregressive and moving average, along with differencing to make the data stationary.

**● Applications:** ARIMA has been used to a great extent in predicting future air quality indices from historic data points. It does well in short-term forecasting for PM2.5, PM10, NO2, and SO2.

**● Challenges:** One of the primary challenges of ARIMA is that the data has to be stationary. If they are not stationary, then the data has to be transformed into a stationary one with methods like differencing, which complicates the modelling process. Also, ARIMA may struggle with non-linearity and interactions among variables.

**SARIMA**

It is an extension of ARIMA and includes seasonal components, so it could work with periodic data. This adds additional parameters in the data for seasonality.

**Applications**: SARIMA can handle seasonal trends in air quality data, and hence, has been applied in order to predict changes in pollutant levels due to seasonal variations in weather patterns and human activities.

**Challenges**: The addition of seasonal parameters adds to the complexity of the model. Careful analysis is required in selecting seasonal parameters, and overfitting can result in case it gets too complex.

**2.1.2 Regression Models**

Regression models are statistical techniques that approximate the relationship between variables. They find extensive use in air quality prediction to determine how various factors impact pollutant concentrations.

**Multiple Linear Regression (MLR)**

**Description:** This is a method of regression used to model the linear relationship between more than one independent variable and a single dependent variable. It uses variables such as temperature, humidity, wind speed, and industrial activities to come up with pollutant levels.

**Applications:** Extensive applications of MLR include the prediction of NO2, SO2, particulate matter, and other pollutant concentrations. It's a very simple, easily understandable model, and thus it's often applied at the beginning of any modeling and analysis.

**Issues:** The key assumption of MLR is linearity, which may be rarely guaranteed in complex environmental systems. It might lose all nonlinear interactions and might get sensitive to multi-collinearity between predictors.

Evaluation Metrics for Air Quality Prediction Models .Several metrics are used to help evaluate models in predicting air quality. This helps in ascertaining the accuracy and reliability of the predictions.

**Mean Absolute Error MAE**

**Description**: MAE is the average magnitude of errors between predicted and actual values without considering their direction.

**Applications**: MAE is easy to understand and is utilized in knowing the accuracy of prediction for models like ARIMA, SARIMA, and MLR.

**2.1.3 Challenges to Air Quality Prediction**

For a long time, as predictive modeling techniques improved, a number of challenges have persisted in the accurate prediction of air quality.

**Data Quality and Availability.**

Construction of a good prediction model requires data of high quality, continuous, and comprehensive. Model performance can be reduced by data gaps, inaccuracies, or limited data.

Some of the ways of improving the quality and availability of data involve the implementation of data cleaning techniques, integration of data from several sources, and improvement of methods of data collection.

**Complexity of the model**

These complex models may capture intricate patterns in the data, but simultaneously, they can suffer from issues related to overfitting that come along with high computational resources. Model performance should be traded off with model complexity.

Model complexity may be controlled using techniques such as cross-validation, regularization, and model simplification. They avoid overfitting.

The factors that influence air quality vary greatly and range from meteorological conditions to industrial activities to the pattern of traffic. The modeling of the interactions among these factors—from the field scale to the regional background and long-range transport—is complex because of the dynamics and interactions involved.

In this regard, real-time meteorological data, ensemble methods, and hybrid models could improve predictive accuracy.

This literature review provides the overall view of the current status of the methods applied for air quality prediction by detailed research of the existing systems and problems associated with them. Only by grasping these systems can more accurate and reliable predictive models be designed in the future.

**2.2 Proposed System**

In this section, an integrated system for the prediction of air quality will be comprehensively proposed, which integrates data collection and its preprocessing with new models of machine learning to improve the accuracy and reliability of the air quality forecast.

**2.2.1. Data Collection**

Firstly, diverse and complete datasets will be acquired from multiple sources to provide the most robust predictive modeling under the proposed system.

**Meteorological Data**

**Variables:** Temperature, speed of wind, relative humidity, pressure, and other meteorological variables should all be included.

**Sources:** This kind of data can be directly received from meteorological stations, weather services, and satellite observations.

**Pollutant Data**

**Variables:** Information on pollutant concentrations of SPM, RSPM, NO2, SO2, O3, CO, etc., has to be collected.

**Sources:**Utilize ground-based sensors, air quality monitoring stations, and satellite remote sensing.

**Emission Data**

**Sources:**Obtain data on emissions from industrial activities, vehicular traffic, and other anthropogenic sources. This data can be sourced from governmental databases, industry reports, and traffic monitoring systems.

**Geographical Data**

**Variables:** Add information on topography, vegetation, land use, and the urban infrastructure.

**Sources:**One can get full geographical information from GIS databases and remote sensing data.

**2.2.2. Data Preprocessing**

Efficient techniques for data preprocessing are keys to providing high-quality and consistent data sets to machine learning models. Major aspects of data preprocessing include data cleaning, transformation, temporal alignment, and spatial interpolation. Data cleaning handles missing values through interpolation or imputation and uses robust statistical methods for outlier detection and handling so that they will not skew model predictions. Filtering methods and other noise reduction techniques are also applied to make the noisy data smooth, hence improving the quality of the dataset.

Another major step will be the transformation of data, where normalization and standardization take place. Normalization rescales the data into a common range so that no analysis is inconsistent, while standardization changes it to zero mean and one standard deviation, again helping in improving the model performance. Temporal alignment synchronizes data from different sources into a standard time frame; hence, temporal consistency across datasets is retained. Spatial interpolation is used to estimate the concentration of pollutants where no direct sensor data exist. Kriging methods complete spatial gaps and thereby provide a more complete and continuous dataset for further modeling. All these preprocessing techniques are applied in ensuring high quality and consistency of the datasets, which are then fed to some machine learning models proposed by the system to obtain more accurate and reliable air quality predictions.

**2.2.3. Machine Learning Models**

In this , different machine learning models have been proposed to be used in predicting air quality indices. These were chosen for their capability to deal with complex datasets and produce accurate predictions. Decision trees are self-descriptive models, intuitive, easy to interpret, and make a decision through an input feature based on a tree-like model. They are suitable for the prediction of air quality indices by splitting data into branches according to the values of features. On top of that, random forests use the ensemble method for multiple decision trees to be aggregated for further accuracy and prevention of overfitting. This algorithm is useful in two ways: first, with huge datasets having several features; and second, it will be able to pick out complex interactions among variables. Support Vector Machines are also applied in the system due to their efficiency in high-dimensional spaces.It uses SVMs that will be applied for regression tasks and in search of the best hyperplane separating data points of feature space, thus able to predict air quality indices in complex datasets. Similarly, K-Nearest Neighbors is used since it is simple yet efficient on small datasets.KNN makes a prediction of the value of any data point by averaging their k- nearest neighbors; it works on the principle that nearby data points should have similar values to make an accurate prediction.

It also provides linear regression, a simple model predicting the dependent variable based on the linear relationship with one or more independent variables, which helps in preliminary analysis and understanding of the linear relationships in data. In classification tasks, logistic regression is predominantly used to calculate the probability of a given input point belonging to a certain class; thus, in this case, it classifies air quality into good, moderate, or poor. Decision tree classifier: It is applied to split the dataset into subsets depending on the feature values; in this case, it is used to classify the levels of air quality and compare the accuracy against other models.

To that end, the testing is supposed to be against the accuracy of the proposed system using the trained dataset. Another important aspect of evaluation metrics will include the Mean Absolute Error, Root Mean Squared Error, and R-squared. All-rounded, the proposed system is to enhance the accuracy and reliability of air quality prediction, hence, consequently offering valuable insights into the management of environmental health risks and formulation of effective pollution control strategies.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Importance of Design**

Air quality prediction is a vital component in the broader effort to mitigate the adverse effects of pollution on human health and the environment. With the rapid urbanization and industrialization seen across the globe, cities are experiencing unprecedented levels of air pollution. This rise in pollution directly impacts respiratory and cardiovascular health,leading to increased morbidity and mortality rates. Therefore, accurately predicting air quality can provide timely warnings to the public and allow for preventive measures, thereby reducing exposure to harmful pollutants.

**3.1.1 Accuracy:**

Accuracy is the most paramount aspect of air quality prediction systems. It is the component that directly influences the reliability of the forecasts and the effectiveness of any resulting actions or recommendations. The system shall have robust algorithms working with high quality data so that the predictions of Air Quality Indices are very close to real conditions, ensuring accurate predictions result in timely alerts and advisories, very critical to public health and safety. For that reason, frequent updates and validations of the predictive models are necessary to maintain high accuracy over time.

**3.1.2 Latency:**

EA real-time air quality monitoring and prediction system must have low latency. A system processing inwardly the sensor data to predict without much delay is very important to ensure timely information to the user. Fast response time is needed for any immediate decisions to be taken, like warnings or corrective measures in case of deterioration in the quality of air. High latency may render the system ineffective and could result in missing an opportunity for intervention or mitigation.

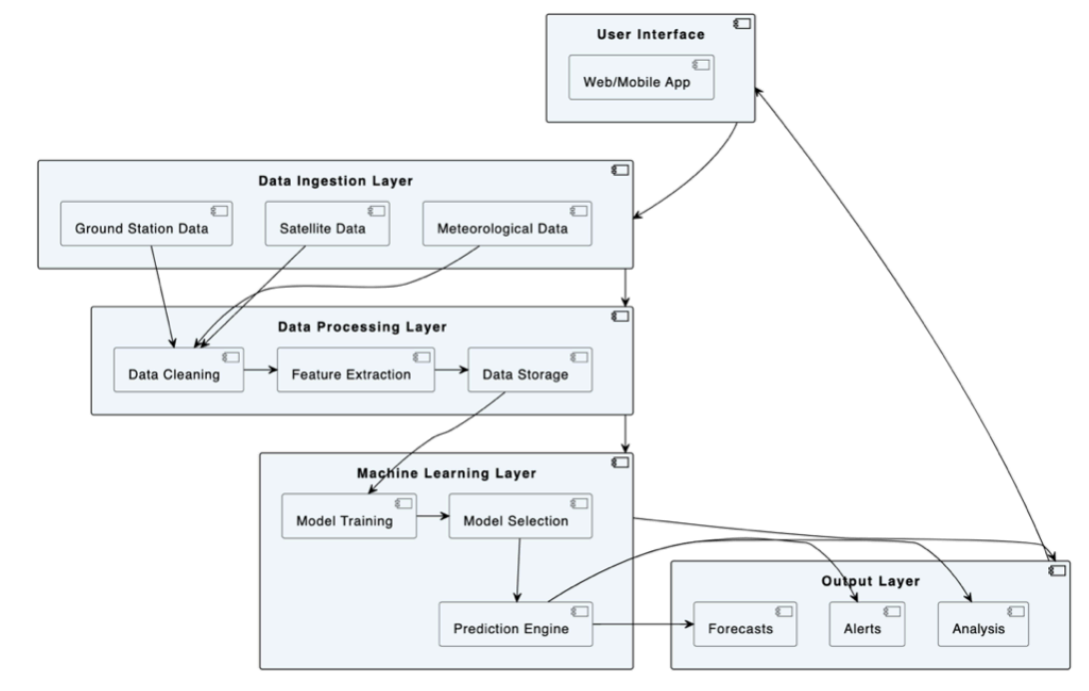
**3.1.3 Scalability:**

The system shall be designed with scalability to allow any volume of data to be generated when the number of sensors and data points increases. The system shall handle and analyze data from multiple sources without loss of performance. Scalability assures flexibility in the system to grow as sensor networks expand and data volumes increase, all with the maintenance of the capability to predict precisely and effectively when demands on the system change.

**Reliability and Robustness:**

● Reliability and robustness are important to ensure continuous and dependable operation of the air quality prediction system. It should be designed to work consistently without too much frequency of failures and have mechanisms that will enable graceful handling of unexpected issues—noisy or incomplete data—to keep running without significant degradation in performance. Robustness concerns the implementation of mechanisms able to identify and correct errors so that the system remains able to work and to make reliable predictions even under hostile environmental or operational conditions.

**3.2 System Architecture**



**Fig.3.1.System Architecture**

1. **Data Collection and Storage**

● **Sources of Data**: Government agencies, environmental monitoring stations, online APIs, historical data repositories.

● **Data Types**: The following would form a part of the dataset: pollutant concentration levels for SO2, NO2, RSPM, SPM, PM2.5; meteorological data like temperature,humidity, wind speed; geographical data like location coordinates.

● **Storage**: Data is stored in a centralized database (e.g., SQL, NoSQL) or data lake (e.g.,AWS S3, HDFS).

1. **Data Preprocessing**

● **Data Cleaning**: Handling missing values, outlier detection, and correction.

● **Feature Engineering**: Calculation of individual pollutant indices (SOi, Noi, RSPMi,SPMi) and Air Quality Index (AQI).

● **Normalization/Standardization**: Scaling numerical features to a standard range

● **Label Encoding**: Converting categorical variables into numerical format.

1. **Exploratory Data Analysis (EDA)**

● **Visualization**: Using libraries like Matplotlib, Seaborn for visualizing pollutant levels across different states and time periods.

* **Statistical Analysis**: Descriptive statistics to understand the distribution and correlation of features.

1. **Model Training**

● **Train-Test Split**: Dividing the dataset into training and testing sets.

● **Model Selection**: Training multiple models such as Linear Regression, Decision Tree Regressor, and Random Forest Regressor for regression tasks, and Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and KNN for classification tasks.

1. **Model Evaluation**

● **Metrics for Regression**: RMSE (Root Mean Squared Error), R2 Score.

● **Metrics for Classification**: Accuracy, Cohen's Kappa Score, Confusion Matrix.

● **Cross-Validation**: K-Fold Cross-Validation to ensure model robustness.

1. **Model Deployment**

● **Model Export**: Saving the trained model using serialization libraries (e.g., Pickle,Joblib).

● **Deployment Environment**: Setting up a server (e.g., Flask, Django) to host the prediction API.

● **Monitoring and Maintenance**: Continuous monitoring of model performance and

retraining with new data as necessary.

1. **User Interface**

● **Dashboard**: Visualization tools (e.g., Plotly, Dash) for real-time monitoring of air quality

**3.2 Functional Requirements**

**Software Requirements**

**1.Real-Time Data Processing**

For effective air quality prediction, the software must be capable of real-time data processing.This involves collecting and analyzing data from sensors as it arrives, without delays that could impact the timeliness of predictions and alerts. Real-time processing ensures that users receive up-to-date information about air quality conditions, which is crucial for timely interventions and decision-making. The software should include robust data ingestion pipelines and efficient processing algorithms to handle continuous streams of sensor data effectively.

1. **Predictive Modeling Capabilities**

The software must support advanced predictive modeling techniques to forecast air quality. This involves integrating machine learning algorithms, statistical models, or other predictive methodologies that can analyze historical and real-time data to generate accurate forecasts of air quality indices. The system should facilitate model training, validation, and updating, ensuring that predictions remain accurate and relevant as new data is incorporated. Flexibility in model selection and the ability to experiment with different algorithms can enhance predictive performance.

1. **Data Integration and Management**

Effective air quality prediction requires seamless integration of data from various sources.The software should support diverse data formats and sources, including different types of sensors and external datasets (e.g., weather data). Efficient data management features are necessary for organizing, storing, and retrieving data. The system should include functionalities for data normalization, aggregation, and cleansing to ensure that the integrated data is accurate and consistent, enabling reliable predictions.

1. **User Interface and Visualization**

A user-friendly interface is essential for presenting air quality predictions and historical data to end-users. The software should offer intuitive dashboards, visualizations, and reporting tools that 18make it easy for users to interpret air quality data and trends. Features such as real-time updates, interactive maps, and alert notifications enhance user engagement and decision-making. The interface should be designed to accommodate different user roles, providing appropriate levels of access and functionality based on user needs.

1. **Security and Privacy**

Security and privacy are critical considerations in the software requirements for air quality prediction. The system must implement robust security measures to protect data from unauthorized access or tampering. This includes user authentication, data encryption, and secure communication protocols. Additionally, the software should comply with relevant privacy regulations to safeguard user information and ensure that data is handled responsibly. Ensuring data integrity and confidentiality is vital for maintaining trust and reliability in the air quality prediction system.

**Hardware Requirements**

1. **Sensor Hardware**

For air quality prediction, high-quality sensor hardware is essential for accurate data collection.Sensors must be capable of measuring various pollutants, such as CO2, PM2.5, NO2, and ozone,with precision and reliability. They should be robust and designed to operate under various environmental conditions to ensure consistent performance. The hardware must also include calibration capabilities to maintain accuracy over time. The choice of sensors can impact the overall data quality and, consequently, the accuracy of the air quality predictions.

1. **Data Acquisition Devices**

Data acquisition devices are necessary to gather and transmit data from sensors to the central processing system. These devices must be capable of interfacing with multiple sensors, collecting data at high frequencies, and transmitting it in real time. They should include features such as data buffering and error handling to ensure reliable data transmission. The hardware should also support various communication protocols (e.g., wireless, wired) to integrate with different types of sensor networks effectively.

1. **Computational Resources**

Computational resources, including servers or cloud infrastructure, are needed to process and analyze the collected data. These resources must have sufficient processing power,memory, and storage to handle large volumes of real-time data and run complex predictive models. Scalability is an important consideration, as the system should be able to expand computational capabilities as data volume and processing demands increase. High-performance CPUs or GPUs may be required for running advanced machine learning algorithms and ensuring efficient data processing.

1. **Networking Infrastructure**

Reliable networking infrastructure is crucial for the seamless transfer of data between sensors, data acquisition devices, and the central processing system. This includes robust networking hardware such as routers, switches, and network interfaces that can handle high data throughput and ensure low latency. The network should be designed to provide stable and secure connections, with redundancy and fail over mechanisms to maintain data integrity and system availability in case of network disruptions.

1. **Power Supply and Backup Systems**

Power supply and backup systems are vital to ensure continuous operation of the air quality prediction hardware. Sensors and data acquisition devices need a stable power source to function correctly. In areas with unreliable power, backup systems such as uninterruptible power supplies (UPS) or generators are necessary to prevent data loss and maintain system operation during power outages. The hardware should also be designed for energy efficiency to minimize power consumption while ensuring reliable performance .

**Memory (RAM):**

A minimum of 8 GB RAM is recommended for handling larger datasets and training complex models. During model training and data processing tasks, RAM is used to store temporary data, intermediate results, and model parameters. With sufficient RAM, the system can efficiently load and manipulate large datasets in memory, reducing the need for frequent disk access and improving overall performance. Additionally, higher RAM capacity allows for smoother multitasking and better responsiveness when working with resource-intensive applications.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Module Description**

**PANDAS**

One of the core libraries that comprise the Python programming language, pandas deals with data manipulation and analysis. At the root of this library lies the so-called DataFrame. It's a generalization of a data structure similar to a spreadsheet or SQL table. This very object, DataFrames, empowers many types of operations to work on tabular data with high performance in store and manipulation. It also offers easy filtering, grouping, and transforming. This is why pandas is so important to data scientists, analysts, and everyone else who works with large datasets.They also provide an easy way to filter, group, and transform. This is the reason why pandas is important to data scientists, analysts, and all those working with large data sets.

Some of the simplest features of pandas include reading and writing data in various formats, like CSV, Excel, and SQL databases. As an example, the function read\_csv() provides seamless importation of any CSV file into a DataFrame. It greatly assists in data preprocessing, as users can get their data into a structured format very fast where they can start doing their analysis. Similarly, the to\_csv() method clearly provides a route to export data back into CSV format once manipulation is complete.

Exploratory analysis is arguably one of the most critical stages when working with data, and pandas provide several methods to make this easier. The head() function returns the first few elements of a DataFrame, thus providing a snapshot of the data. The info() method provides a decent view of what the structure of the DataFrame looks like, including how many non-null entries are in each column and what their data type is. For statistical summaries, the describe() function builds descriptive statistics, such as mean, standard deviation, and percentiles, which are characteristics important for understanding the distribution of variables.

Another critical part of data preparation is cleaning the data. In this domain, pandas has very powerful tools. For example, fillna() enables replacement of missing values with the specified values or interpolation methods. Then, there is dropna(), which drops rows or columns with missing data. Again, there is the drop() method, which comes in handy in getting rid of unimportant columns or rows, thereby streamlining this DataFrame to just those of relevance for analysis.

Beyond cleaning, pandas has robust methods for data manipulation. There are functions for group-true operations and aggregations needed to compute the averages or sums for different categories using groupby(). Next, there are merge() and concat() functions that combine several DataFrames into one. All these features make it possible to do complex data integration. Thus, pandas is a full-scale package to transform raw data into meaningful insights and hence drive home the importance of its place in the workflow of data analysis.

**NUMPY**

Numpy is a library with extremely versatile and efficient processing of arrays, proposing various tools to work with multidimensional arrays. This library's main application is scientific computing with Python. Many functionalities of a basic order are contained within this package; therefore, when working with data analysis, it is a core element.

Among the main features of the Numpy package in working with arrays is the high performance for the manipulation of N-dimensional arrays. The ease of working over large datasets makes this tool ideal for multi-dimensional array operations required for the analysis tasks.

Numpy is also equipped with very sophisticated broadcasting capabilities that make computations and manipulations on arrays quite efficient. This basically refers to how Numpy handles operations between two arrays of different shapes and sizes. It is pretty easy to do through the broadcasting features in numpy. Apart from its functions in manipulating arrays, Numpy enables the implementation of codes written in languages like C/C++ and Fortran. The utility is quite important in scientific computing, particularly where any external libraries or modules are required for the task.

Numpy also has an enormous set of mathematical functions at one's disposal; many of them are useful, particularly the linear algebra and Fourier transform tools and random number generators. That is what makes it quite a versatile tool for a great variety of scientific computing tasks.

It can also be used to be an ultra-efficient and flexible container to store and manipulate nonspecific data. Besides, using Numpy, one can define arbitrary data types that are very fast and easily incorporated into an array of databases.

In general, Numpy is very helpful in scientific computing and data analysis because of the simple fact that the number of features and capabilities let a user efficiently work on large datasets.

**SEABORN**

Seaborn is a Python library built on top of Matplotlib that offers robust data visualization.It makes creating complex, nice-looking plots very fast with just very few lines of code. It has been developed to provide a high-level interface that is closely integrated with the pandas DataFrame and intimately connected with the matplotlib library. Seaborn is very good at statistical graphs, which are most useful in understanding relationships between variables and sometimes finding patterns within datasets.

One of the main advantages of Seaborn over other visualization libraries is in creating pair plots. This plot comes in very handy when one wants to see relationships between multiple variables at one go. It creates a matrix of scatterplots for each pair of variables, which gives an exhaustive insight into how variables interact and possibly correlate with one another. Besides, Seaborn bar plots are an explicit and neat way to visualize categorical data, for example, how pollutants are distributed across states. With different in-built themes and color palettes that Seaborn has, one can quite easily dress up the plot to make it not only informative but very engaging to the eye. This marriage of functionality with style makes Seaborn a very useful tool for data analysis and its presentation.

**MATPLOTLIB**

Matplot Lib is one of the most versatile and widely used plotting libraries in Python,enabling the generation of static, animated, and interactive plots. Developed by John D. Hunter, the earliest version was released in 2003; today, it supports a very large framework of methods and tools for data visualization in most formats. This library is argued to be highly flexible in making a plot look the way one wants because it allows users to control nearly anything in the visualization, such as figure size, resolution, color schemes, and line styles. This level of control makes it an indispensable tool for data scientists, analysts, and researchers who have to communicate their results effectively.

Some of the core strengths of Matplotlib relate to creating different types of plots and charts.It is possible to create histograms, bar plots, scatter plots, and line graphs. Each plot type can be tailored to meet the unique needs of analysis. For example, it is possible to get histograms that use different bin sizes or ranges; different colors, patterns, and spacing are possible in bar plots. Further, Matplotlib supports annotations, legends, and gridlines, all of which enable a plot to be more interpretable. This flexibility is one of the reasons it is an ideal library to create lucid and informative visual representations of complicated data.

Matplotlib also excels in plot aesthetics and layout management. Users can set the figure size, adjust the aspect ratio, and rotate labels to ensure that plots are both visually appealing and easy to read. The library supports a range of styles and themes, allowing users to match the look and feel of their plots to specific presentation or publication requirements. Furthermore, Matplotlib’s integration with other libraries such as NumPy and Pandas facilitates seamless data handling and manipulation, making it a cornerstone of the Python data visualization ecosystem. Whether for exploratory data analysis or formal reporting, Matplotlib provides the tools needed to create high-quality and effective visualizations.

**SKLEARN**

The sklearn library is a robust toolkit for machine learning in Python, containing a large variety of tools for preprocessing and model selection and model evaluation. Sklearn.preprocessing is a module of utmost importance in the process of setting up raw data for modeling, as it implements a number of preprocessing techniques. For example, LabelEncoder will be applied to transform the categorical labels into numerical format since many algorithms of machine learning work with numerical inputs. This module also contains other preprocessing tools, including StandardScaler for normalizing the values of features, and OneHotEncoder for transforming categorical features into a binary matrix.

The sklearn.model\_selection module is a vital way of cutting your data into training and test sets. This is good practice to check model performance and to avoid overfitting. Too large a variance makes a model unapt at generalizing well to new data. Functions like train\_test\_split help to create those splits, ensuring that the model's performance can be tested on unknown data. In this module there are more tools for cross-validation, which helps in reliable assessment of the model by systematic division of data into many different training and validation sets.

It provides a number of linear algorithms for regression and classification modeling in sklearn.linear\_model. The LinearRegression class is used to model the relationship between features and target variables; this would be applied in predicting the air quality index with respect to other input features. In contrast, decision tree algorithms, such as DecisionTreeRegressor and DecisionTreeClassifier, can be found in sklearn.tree for regression and classification tasks correspondingly. These trees are used to split the data on feature values to predict the outcome or classify data into different categories. In an effort to make the model more accurate, sklearn.ensemble proposes ensemble methods such as RandomForestRegressor and RandomForestClassifier, which merge multiple decision trees to improve performance and robustness. Finally, the metrics module available in sklearn.metrics has a number of functions that measure model performance, among them Mean Absolute Error, Root Mean Squared Error, and accuracy scores. These metrics are mostly used when assessing how good the predictive model is and if the set performance standard has been met. Additionally, sklearn.neighbors also include KNeighborsClassifier classifying points based on the proximity of their nearest neighbors; therefore, making predictions by using the k-nearest neighbors algorithm gives similarity in data.

In a nutshell, Scikit-learn is an end-to-end library that caters to the needs of an entire machine learning pipeline, starting from preprocessing data and model selection to the evaluation of their performance. There are various modules and functions that make it possible for users to build, train, and evaluate most kinds of machine learning models efficiently.

**WARNINGS**

This module allows the author to control and handle warnings thrown by the Python interpreter. In the code, warnings.filterwarnings("ignore") is applied for the suppression of warnings that might clutter the output so that it remains clean and focused on results during the code run.

**4.2 Module Components**

**Python:**

Python is a multi-purpose programming language that is readable and easy to use. Due to its rich variety of libraries and frameworks, it is very widely applied in Data Science and Machine Learning. Python will be used in the given code for data manipulation,application of Machine Learning algorithms, and finally, visualization. It has a broad ecosystem of libraries, like pandas, numpy, scikit-learn, and visualization tools including matplotlib and seaborn, making it well suited for carrying out all kinds of analyses on data and model development.

**4.2.1 Dataset**

**cleaned\_air\_quality\_dataset.csv:**

SNo.,Date,City,PM2.5,PM10,NO,NO2,NOx,NH3,CO,SO2,O3,Benzene,Toluene,Xylene,AQI,AQI\_Bucket this parameters given in the dataset which is used to analyse the output as per the values in the dataset.

**Table. 1.dataset explanation table**

| **Parameter** | **Description** |
| --- | --- |
| **SNo.** | Serial number (index, not useful for modeling) |
| **Date** | Date of observation |
| **City** | City name |
| **PM2.5** | Fine particulate matter (diameter ≤2.5 µm) |
| **PM10** | Coarse particulate matter (diameter ≤10 µm) |
| **NO** | Nitric Oxide |
| **NO2** | Nitrogen Dioxide |
| **NOx** | Nitrogen Oxides |
| **NH3** | Ammonia |
| **CO** | Carbon Monoxide |
| **SO2** | Sulfur Dioxide |
| **O3** | Ozone |
| **Benzene** | Volatile organic compound |
| **Toluene** | Volatile organic compound |
| **Xylene** | Volatile organic compound |
| **AQI** | Air Quality Index (numeric value) |
| **AQI\_Bucket** | Categorical air quality level (e.g., Good, Satisfactory, Moderate, Poor, Very Poor, Severe) |

**aqi\_dataset.csv**

City,Date,PM2.5,PM10,NO,NO2,NOx,NH3,CO,SO2,O3,Benzene,Toluene,Xylene,AQI,AQI\_Bucket this parameters given in the dataset which is used to analyse the output as per the values in the dataset.

**Table. 2.dataset explanation table**

| **Type** | **Columns** |
| --- | --- |
| **Identification** | City, Date |
| **Input Features** | PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, Xylene |
| **Target Output** | AQI, AQI\_Bucket |

**Table. 3.dataset explanation table**

| **Analysis Type** | **Purpose** |
| --- | --- |
| **Descriptive Statistics** | Average values, distribution of pollutants, AQI range |
| **Correlation Analysis** | Identify which pollutants are most strongly linked to AQI |
| **Regression Modeling** | Predict AQI from pollutant levels |
| **Classification Modeling** | Predict AQI\_Bucket from pollutant levels |
| **Trend Analysis** | Explore how AQI/pollutants change over time or across cities |
| **City Ranking** | Rank cities based on average AQI or pollution levels |
| **Feature Importance** | Discover which pollutants most affect air quality |

**4.2.2 Data Analysis and Visualization**

**Data Analysis :** These are technologies concerned with the preparation and refining of data into a form fit for analysis. There are some key steps involved in this stage,including cleaning the data, whereby there will be imputation of missing values and the treatment of outliers; the inconsistencies will also be resolved. For example, missing data can be handled by simple mean imputation or by multiple imputation methods. Another critical aspect is data transformation. Depending on the methods used, such as normalization and standardization, it ensures that each of the features is on equal footing during the analysis, especially in algorithms sensitive to feature scaling. In addition, joining datasets, restructuring data, encoding categorical variables, among others, are also techniques included in data wrangling. All these processes work toward making data understandable and ready for use. Effective manipulation of data not only ensures that one has the right data set but, more importantly, in the best format for the following steps in analysis and model building.

**Data Visualization:**

Broad adaptations to tools and libraries, such as Seaborn, Matplotlib, and Plotly, are truly important in bringing across complex data into clear, meaningful,and actionable insights. In view of the foregoing, effective visualization can therefore reveal many hidden patterns, trends, and correlations in data that might otherwise not be obvious in the raw format. Scatter plots can represent, for instance, the relationships between variables. Heat maps are designed for graphical representation of density and distribution of data. A data scientist and an analyst can, therefore, present their findings using visualization in a more self-explanatory and pleasing manner that allows for better decision-making and communication of insights to various categories of stakeholders.

In machine learning, model development and evaluation follow a systematic process of building predictive models using a host of available algorithms whereby they are then evaluated. This will involve the development of models based on appropriate algorithms against tasks — for instance, linear regression in the case of continuous outcomes, or decision trees for classification tasks.It trains the model using historical data, tunes hyper-parameters to attain maximum performance, and validates the model using various techniques such as cross-validation to check the goodness of its fit over unseen data.During this model evaluation step, different evaluation metrics benchmarking performance models for iterative improvements would be used, including accuracy, precision, recall, and F1-score. This is an extremely critical step in the development of robust models that perform well during training and for the reliable provision of predictions in real-world situations.

**Frameworks:**

It is a very powerful, flexible, and useful machine learning framework called scikit-learn. It offers numerous tools to construct and evaluate a large variety of predictive models. Normally, scikit-learn modularizes different tasks—for example, preprocessing, training a model, or evaluating its performance—to make a machine learning workflow easier. Skikit-learn is a superclass to many algorithms for classification, regression, clustering, and dimensionality reduction; hence, it is appropriate for most of the different machine learning problems. Its friendly API and great documentation make this superclass very approachable and easy to realize these complex techniques of machine learning within Python. Thanks to its modular architecture, this framework can easily be integrated into pipelines for data analysis and its power harnessed in preprocessing data, building models, and evaluating their performance in a coherent, friendly environment.

**4.3 Sample Code**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.tsa.arima.model import ARIMA

import numpy as np

import torch

import torch.nn as nn

from sklearn.preprocessing import MinMaxScaler

df = pd.read\_csv('aqi\_dataset.csv')

print(df.head())

print(df.info())

print(df.columns)

df["Date"] = pd.to\_datetime(df["Date"])

#DATA CLEANING

print("Null values in each column:")

print(df.isnull().sum())

print("Duplicate rows:", df.duplicated().sum())

df = df.drop\_duplicates()

df = df.dropna()

print(f"Original dataset shape: {df.shape}")

print(f"Cleaned dataset shape: {df.shape}")

data\_filled = df.fillna(0)

print("Null values after cleaning:")

print(df.isnull().sum())

#feature engineering

highest\_aqi\_date = df.groupby('Date')['AQI'].mean().idxmax()

highest\_aqi\_value = df.groupby('Date')['AQI'].mean().max()

print(f"The date with the highest average AQI is {highest\_aqi\_date} with an AQI of {highest\_aqi\_value:.2f}")

city\_highest\_pm25 = df.groupby('City')['PM2.5'].mean().idxmax()

highest\_pm25\_value = df.groupby('City')['PM2.5'].mean().max()

print(f"The city with the highest average PM2.5 is {city\_highest\_pm25} with PM2.5 concentration of {highest\_pm25\_value:.2f}")

pollutants=['PM2.5','PM10','NO','NO2','NOx','NH3','CO','SO2','O3','Benzene','Toluene','Xylene']

pollutants\_avg = df[pollutants].mean()

print("Overall Average Values for Key Pollutants:")

print(pollutants\_avg)

severe\_aqi\_days = (df['AQI'] > 300).sum()

print(f"The number of days with AQI greater than 300 (severe) is {severe\_aqi\_days}.")

city\_aqi\_variance = df.groupby('City')['AQI'].var().sort\_values(ascending=False).head(5)

print("Top 5 Cities with Maximum AQI Variance:")

print(city\_aqi\_variance)

#EDA

plt.figure(figsize=(10,6))

sns.histplot(df['AQI'], bins=30, kde=True, color='skyblue')

plt.title('Distribution of Air Quality Index (AQI')

plt.xlabel('AQI')

plt.ylabel('Frequency')

plt.show()

city\_aqi = df.groupby('City')['AQI'].mean().sort\_values(ascending=False).head(10)

plt.figure(figsize=(12,6))

city\_aqi.plot(kind='bar',color='coral')

plt.title('Top 10 Cities with the highest average AQI')

plt.xlabel('City')

plt.ylabel('Average AQI')

plt.xticks(rotation=45)

plt.show()

plt.figure(figsize=(12,6))

df.groupby('Date')['AQI'].mean().plot(color='red')

plt.title('AQI Trend Over Time')

plt.xlabel('Date')

plt.ylabel('Average AQI')

plt.show()

plt.figure(figsize=(12,6))

sns.kdeplot(df['PM2.5'], label='PM2.5', color='blue', fill=True)

sns.kdeplot(df['PM10'], label='PM10', color='orange', fill=True)

plt.title('Comparison of PM2.5 and PM10 Levels')

plt.xlabel('Concentration')

plt.ylabel('Density')

plt.legend()

plt.show()

plt.figure(figsize=(8, 6))

sns.countplot(x='AQI\_Bucket', data=df, palette='viridis')

plt.title('Distribution of AQI Buckets')

plt.xlabel('AQI Bucket')

plt.ylabel('Count')

plt.show()

cor\_map = df[['PM2.5','PM10','AQI']]

plt.figure(figsize=(10,8))

sns.heatmap(cor\_map.corr(),annot=True,cmap='coolwarm')

plt.title('correlation heatmap')

plt.show()

# ARIMA Forecast (next 30 days)

df = df.sort\_values('Date')

df.set\_index('Date', inplace=True)

aqi\_series = df['AQI']

from statsmodels.tsa.arima.model import ARIMA

aqi\_last100 = aqi\_series[-100:]

model = ARIMA(aqi\_last100, order=(2,1,2))

model\_fit = model.fit()

forecast\_arima = model\_fit.forecast(steps=30)

last\_date = aqi\_last100.index[-1]

future\_dates = pd.date\_range(start=last\_date + pd.Timedelta(days=1), periods=30, freq='D')

plt.figure(figsize=(12,6))

plt.plot(aqi\_last100.index, aqi\_last100, label='Actual AQI (Last 100 Days)')

plt.plot(future\_dates, forecast\_arima, label='ARIMA Forecast (Next 30 Days)', linestyle='--')

plt.title('AQI Forecast using ARIMA')

plt.xlabel('Date')

plt.ylabel('AQI')

plt.legend()

plt.show()

#PyTorch LSTM forecast

aqi\_values = aqi\_series.values.reshape(-1, 1)

scaler = MinMaxScaler()

aqi\_scaled = scaler.fit\_transform(aqi\_values)

look\_back = 10

train\_lstm = aqi\_scaled[-100:]

def create\_sequences\_torch(data, look\_back):

X, y = [], []

for i in range(len(data) - look\_back):

X.append(data[i:i+look\_back, 0])

y.append(data[i+look\_back, 0])

return np.array(X), np.array(y)

X\_lstm, y\_lstm = create\_sequences\_torch(train\_lstm, look\_back)

X\_lstm = torch.tensor(X\_lstm, dtype=torch.float32).unsqueeze(-1) # shape: (samples, look\_back, 1)

y\_lstm = torch.tensor(y\_lstm, dtype=torch.float32).unsqueeze(-1) # shape: (samples, 1)

class LSTMModel(nn.Module):

def \_init\_(self, input\_size=1, hidden\_size=50, num\_layers=1):

super(LSTMModel, self).\_init\_()

self.lstm = nn.LSTM(input\_size, hidden\_size, num\_layers, batch\_first=True)

self.fc = nn.Linear(hidden\_size, 1)

def forward(self, x):

out, \_ = self.lstm(x)

out = self.fc(out[:, -1, :])

return out

model = LSTMModel()

criterion = nn.MSELoss()

optimizer = torch.optim.Adam(model.parameters(), lr=0.01)

epochs = 50

for epoch in range(epochs):

model.train()

optimizer.zero\_grad()

output = model(X\_lstm)

loss = criterion(output, y\_lstm)

loss.backward()

optimizer.step()

model.eval()

forecast\_lstm = []

input\_seq = train\_lstm[-look\_back:].reshape(1, look\_back, 1)

input\_seq = torch.tensor(input\_seq, dtype=torch.float32)

for \_ in range(30):

with torch.no\_grad():

next\_val = model(input\_seq).item()

forecast\_lstm.append(next\_val)

next\_input = torch.tensor([[[next\_val]]], dtype=torch.float32)

input\_seq = torch.cat((input\_seq[:, 1:, :], next\_input), dim=1)

forecast\_lstm = scaler.inverse\_transform(np.array(forecast\_lstm).reshape(-1,1)).flatten()

plt.figure(figsize=(12,6))

plt.plot(aqi\_last100.index, scaler.inverse\_transform(train\_lstm).flatten(), label='Actual AQI (Last 100 Days)')

plt.plot(future\_dates, forecast\_lstm, label='PyTorch LSTM Forecast (Next 30 Days)', linestyle='--')

plt.title('AQI Forecast using PyTorch LSTM')

plt.xlabel('Date')

plt.ylabel('AQI')

plt.legend()

plt.show()

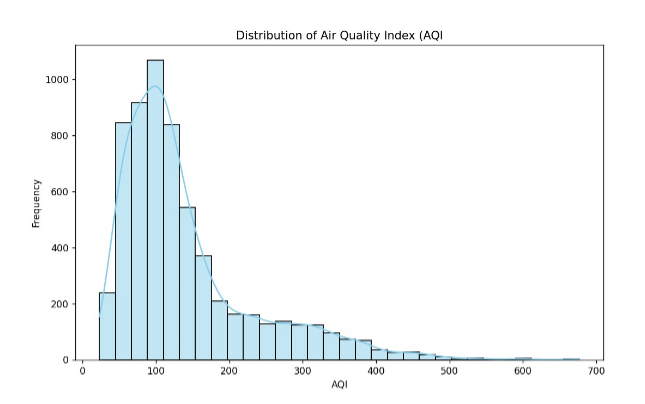
df.reset\_index(inplace=True)

print(df.columns)

SAVE\_DF = df.to\_csv('cleaned\_air\_quality\_dataset.csv',index=True)

**CHAPTER 5**

**RESULTS**

** Fig.2.Distribution of Air Quality Index (AQI)**

The histogram and KDE (Kernel Density Estimation) plot titled "Distribution of Air Quality Index (AQI)" illustrates the frequency distribution of AQI values in a dataset. Here's a brief description and analysis:

**Description:**

X-axis (AQI): Represents the Air Quality Index values, ranging from 0 to about 700.

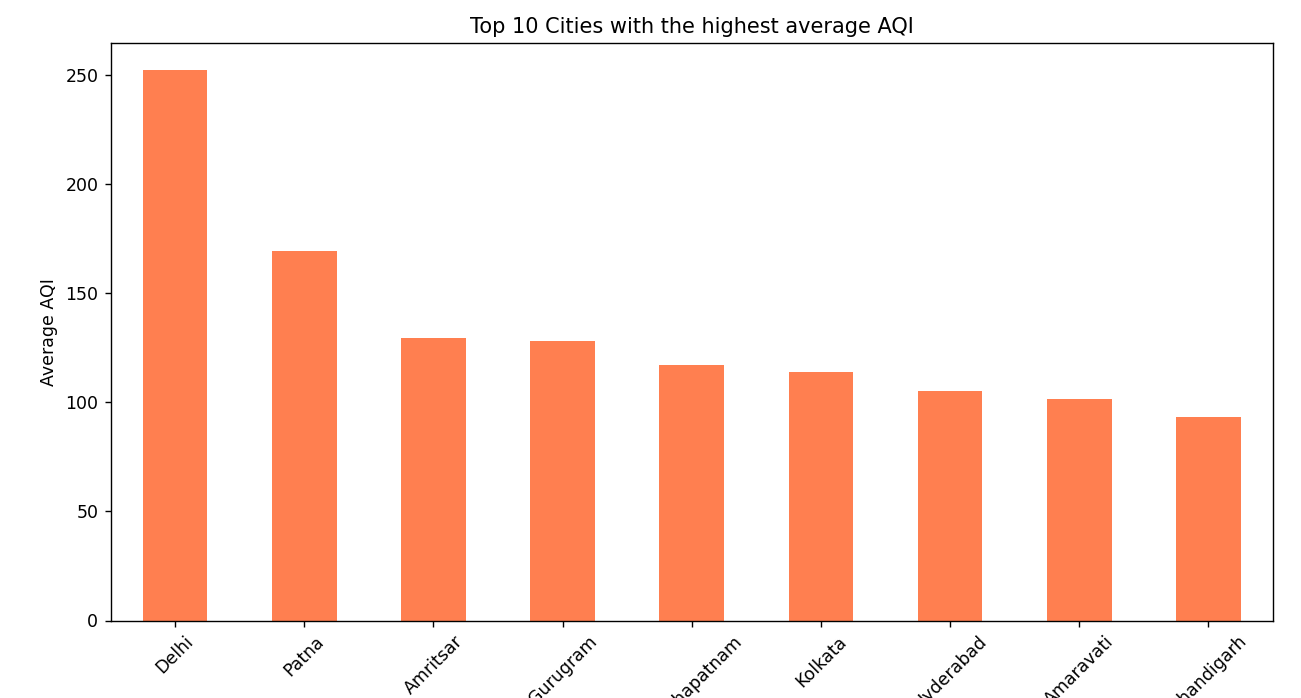
Y-axis (Frequency): Shows how often each AQI range appears in the dataset.

Bars:Histogram bars are colored light blue and show the frequency of AQI values in different bins.

Line:A smooth KDE curve is overlaid to provide a continuous estimation of the AQI distribution.

**Insights:**

The distribution is right-skewed (positively skewed), meaning most AQI values are concentrated on the lower end (good to moderate air quality).The peak occurs around an AQI of 100, indicating this is the most common value range in the dataset.There are fewer instances of very high AQI values (above 300), which are typically considered hazardous.



**Fig.3.Top 10 Cities with the highest average AQI**

The bar chart titled "Top 10 Cities with the highest average AQI" presents a comparison of average air quality index (AQI) values across ten Indian cities. Here's a detailed description and interpretation:

**Chart Breakdown**

X-axis (Cities): Names of the top 10 cities with the highest average AQI.

Y-axis (Average AQI): Average AQI values, indicating overall air quality levels.

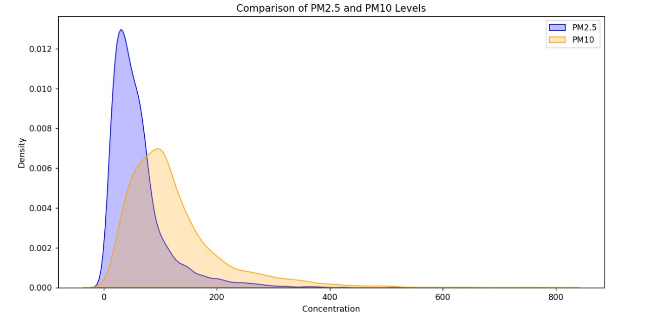
Bars: Each bar represents a city and is shaded in orange, indicating the average AQI.

**Key Observations**

Delhi stands out with the highest average AQI, exceeding 250, indicating very poor to hazardous air quality.Patna is second, with an average AQI around 170, also falling in the unhealthy category.Amritsar and Gurugram follow, both around the 130 mark, which is unhealthy for sensitive groups.Visakhapatnam, Kolkata, Hyderabad, Amaravati, and Chandigarh have AQI values between 90 to 120, still concerning but relatively better compared to Delhi and Patna.

**Health & Policy Implications**

Delhi and Patna may need urgent policy intervention for air pollution control.Even the cities lower on the list are not in the “Good” AQI category, indicating a broader urban air quality issue.



**Fig.4.Comparing the distributions of PM2.5 and PM10**

The image is a kernel density plot comparing the distributions of PM2.5 and PM10 concentration levels in the air.

**What the Plot Shows**

The x-axis represents concentration levels (likely in µg/m³).

The y-axis represents density, a smoothed estimate of the probability distribution.

Blue shaded curve: PM2.5.Orange shaded curve: PM10

**Observations**

1. PM2.5 peaks earlier: The PM2.5 curve has a sharper, narrower peak around a lower concentration (~40–60 µg/m³).This indicates that PM2.5 values are generally lower and more concentrated around a central mean.

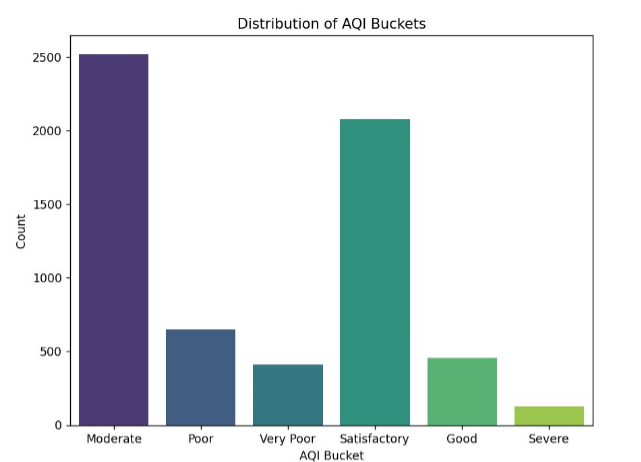
2. PM10 has a broader distribution: The PM10 curve peaks later and extends further right, showing more variability and higher concentrations.This means PM10 levels can often reach higher extremes compared to PM2.5.

3. Long tail for PM10: PM10 values spread across a wider range (up to 800 µg/m³), suggesting that in some instances, PM10 pollution is significantly higher than PM2.5.

**Insights**

Health Implication: While PM10 is often present in higher amounts, PM2.5 is more dangerous due to its ability to penetrate deeper into the lungs and bloodstream.Environmental Insight: PM10 might include coarser particles like dust and pollen, while PM2.5 includes finer particles like combustion residues.

Policy Impact: Effective air quality management must address both PM types, with stronger focus on controlling PM2.5 due to its greater health risks despite lower concentrations.



**Fig.5.distribution of Air Quality Index (AQI) Buckets**

This bar chart shows the distribution of Air Quality Index (AQI) Buckets based on the count of observations in each category.

**AQI Bucket Categories**

These buckets are standard AQI categories (likely based on CPCB or WHO guidelines):

Good,Satisfactory,Moderate,Poor,Very Poor,Severe

**Observations**

1. Most Frequent Category:

Moderate AQI has the highest count, with over 2500 observations, indicating this is the most common air quality level in the dataset.

2.Satisfactory air is the second most common, with over 2000 instances — a positive sign.

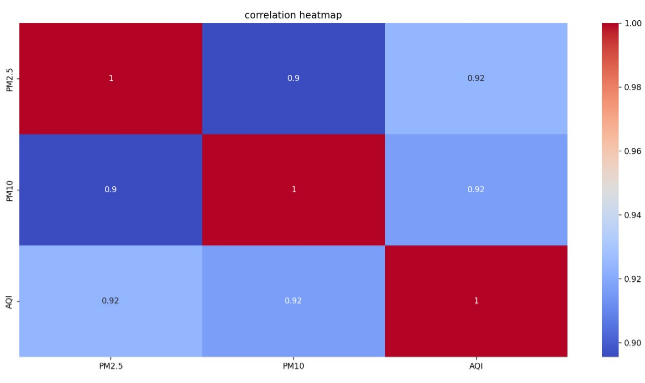
3.Poor and Very Poor AQI levels are less frequent, each with a few hundred observations.

4.Good air quality appears in relatively few instances (less than 500).

5.Severe AQI is the least common, suggesting extreme pollution events are relatively rare in the dataset.

**Insights**

The air quality is most commonly moderate to satisfactory, suggesting that while pollution exists, it’s not overwhelmingly hazardous in most cases.Very few observations are in the 'Good' or 'Severe' extremes — indicating a narrow fluctuation range for AQI across most areas/times.Policy Focus should aim at converting 'Moderate' areas into 'Satisfactory' or 'Good' through interventions in industrial emissions, vehicle control, and green spaces.

****

**Fig.6.Correlation HeatMap of PM2.5,PM10, AQI**

The image is a correlation heatmap showing the relationships between three air quality variables: PM2.5, PM10, and AQI. Here's a full breakdown:

What is a Correlation Heatmap?

A correlation heatmap visually represents the Pearson correlation coefficient between pairs of variables, which ranges from:

+1: Perfect positive correlation,0: No correlation,–1: Perfect negative correlation

Interpretation of the Heatmap

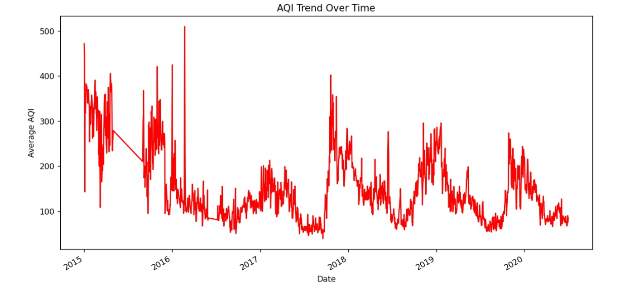
PM2.5 PM10 AQI

PM2.5 1.00 0.90 0.92

PM10 0.90 1.00 0.92

AQI 0.92 0.92 1.00

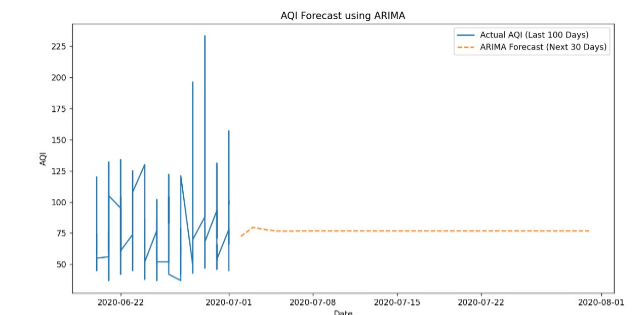
PM2.5 and AQI (0.92): Very strong positive correlation. This suggests that as PM2.5 levels increase, AQI tends to worsen significantly.PM10 and AQI (0.92): Also very strong positive correlation, nearly identical to PM2.5’s influencePM2.5 and PM10 (0.90): High positive correlation, indicating these two particulate matter indicators often rise together.PM2.5 and PM10 are both major contributors to poor air quality, as evidenced by their high correlation with AQI.Policy interventions aimed at reducing particulate matter emissions could significantly improve AQI levels.This analysis reinforces the importance of monitoring fine particulates in air quality assessment and forecasting.



**Fig.7.AQI Trend Over Time as per date and Average AQI**

\*Graph Title:\* AQI Trend Over Time

The line chart visualizes the trend of average Air Quality Index (AQI) over a period from 2015 to 2020. The x-axis represents the date, while the y-axis shows the average AQI values. The line is drawn in red, highlighting changes in air quality levels over time.2015–2016:High fluctuation in AQI, with values frequently exceeding 300, indicating hazardous air quality levels.2017–2018:A noticeable decline in extreme spikes, with AQI generally below 200 but still showing seasonal variations.2018–2020:Recurring peaks suggest periodic pollution surges (likely seasonal), but an overall downward trend in AQI values is observed.Post-2019:The AQI appears to stabilize and decline slightly, possibly indicating improved air quality or effects of environmental regulations.The graph suggests significant air quality improvements over time, despite seasonal spikes. These trends could be attributed to policy changes, public awareness, or external events (e.g., lockdowns, emission controls).



**Fig.8.AQI Forecast Using ARIMA**

Graph Title:AQI Forecast using ARIMA

The line graph visualizes the Air Quality Index (AQI) over time using ARIMA modeling. It includes two segments:

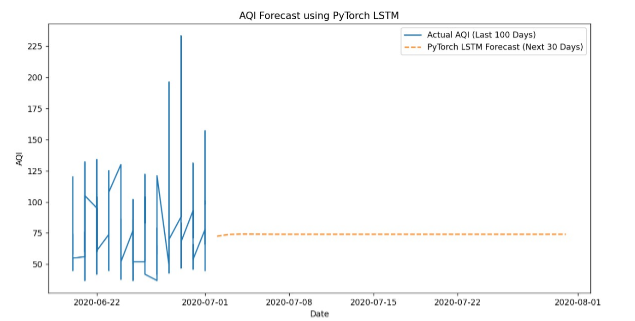
Blue Line (Actual AQI - Last 100 Days):\* This line represents the observed AQI values over the past 100 days. The values show considerable fluctuations, indicating volatility in air quality during this period.

Orange Dashed Line (ARIMA Forecast - Next 30 Days):\* This line represents the forecasted AQI for the upcoming 30 days, predicted using the ARIMA model. The forecast appears to stabilize around an AQI value of \~75, suggesting a consistent air quality level in the short-term future.

X-Axis: Dates from late June to early August 2020.

Y-Axis:AQI values ranging roughly from 40 to 230.

Despite the high variability in historical AQI, the ARIMA model predicts a steady trend in AQI, likely indicating expected environmental stability in the forecasted period.



**Fig.9.AQI Forecast using PyTorch LSTM**

Graph Title:\* AQI Forecast using PyTorch LSTM

This line graph illustrates the Air Quality Index (AQI) trends using a deep learning-based forecasting model, PyTorch LSTM. It consists of two key components:

Blue Line (Actual AQI - Last 100 Days):Shows the real AQI data over the past 100 days. The values exhibit significant fluctuations, with some sharp peaks, indicating inconsistent air quality during this period.

Orange Dashed Line (LSTM Forecast - Next 30 Days): Represents the predicted AQI values for the next 30 days using the PyTorch LSTM model. The forecast indicates a stabilization of AQI around a value close to 75, showing minimal variation over the forecasted period.

X-Axis:Timeline from June 2020 to August 2020

Y-Axis: AQI values, ranging from approximately 40 to over 225

Despite the highly variable historical AQI, the LSTM model forecasts a steady and moderate AQI level in the near future. This implies that the model expects the air quality to remain relatively stable in the coming month.

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

**Conclusion**

This project on air quality prediction is a key milestone in understanding and predicting the levels of air pollution using state-of-the-art data analysis and machine learning techniques. Mainly, it was aimed to develop a complex model that could be used in predicting the air quality indices with high accuracy from a wide base of sources. This work comes at a time when increasing air pollution is a serious threat to human healthand the environment. The purpose of the project was to provide an integrated tool for handling and reducing the impacts of poor air quality.

Firstly, in this project, data was gathered by carefully considering sources of meteorological variables, pollutant concentrations, emission sources, and geographical information. All of the data types are important in understanding the drivers of air quality. Meteorological data—temperature, wind speed, humidity—was combined with pollutant levels for NO2, SO2, particulate matter, emission data from industrial and vehicular sources, and geographical information on topography and land use. The construction of a robust predictive model demanded such a multi-faceted dataset.

Preprocessing of data was that step in the process critical to ensure high quality and reliable information that would then be worked on by models of machine learning. In this project, different methods were used to clean and transform data, handle missing values by interpolation and imputation, and handle outliers and noise by robust statistical methods. This comprised normalization and standardization to ensure consistent scaling and temporal alignment to the same timeframe for data from different sources. Spatial interpolation techniques such as Kriging are used for estimating pollutant levels at places where direct sensor data is lacking. These steps provided a firm base for model training and validation.

This project entailed the application of a good number of machine learning algorithms on the preprocessed data in order to come up with air quality indices. Decision Trees, Random Forest, Support Vector Machines, K-Nearest Neighbors, Linear Regression, and Logistic Regression were in use. Every algorithm was applied according to its strengths dealing with different aspects of the data. For instance, Random Forests and SVMs worked very nicely when handling complex relationships with high-dimensional data, while the simpler models, like Linear Regression and KNN, provided baseline performance metrics. Comparing these models would provide full coverage of their effectiveness in predicting air quality.

Comparing the machine learning models revealed important facts about their performance and applicability. Random forests and SVMs have been very efficient in predicting air quality indices, as they turned out to be more accurate and able to capture complex patterns in data.On the other hand, less complex models like linear regression and KNN supplied a baseline of good performance and insights into linear and distance-based relationships. Some of the evaluation metrics used in this study included the Mean Absolute Error, the Root Mean Squared Error, and R-squared values. All these gave a clear view of each model's strengths and weaknesses.Comparative analysis was very important to select the most appropriate model to be used for the prediction of air quality.

The fact that these models were developed successfully may be interpreted as meaning that machine learning has the capability to improve monitoring and prediction of air quality. These models provide a very good piece of information, especially in the provision of accurate forecasts of air quality indices for policymakers, environment agencies, and the general public. These findings can result in better decisions regarding pollution control measures, health advisories, and environmental policies. Precise predictions of air quality would be a great stride forward in reducing the health risks associated with air pollution and improving general environmental quality.

Looking ahead, there are a myriad of opportunities for improving and developing predictive models. The responsiveness and accuracy of the models can be increased by using future real-time data. Increased models that cover more pollutants and geographical regions would also be very beneficial in giving the overall picture of the dynamics of the quality of the air. Integrating more advanced techniques in such fields as deep learning and ensemble methods can also provide better performance. The overall contribution to the environmental sciences by the project and the facilitation towards the solutions to the present global challenges posed by machine learning are huge Pedestrian.

**Future Scope**

Future Scope of the Pedestrian Signal Analysis System Project.

Having a project based on the effectiveness of the pedestrian signal system, the PSAS project has a great scope for future developments and extended applications. Currently, the pressure on developing intelligent traffic management for changing and growing urban environments is increasing with the fever. A few potential future expansions of this project are considered below.

**1.Integration of Real-Time Data Sources**

One of the future scopes of air quality prediction projects is to integrate real-time data sources to predict air quality more accurately and from a shorter distance. Presently, the models utilize historical data and make periodic updates, but real-time data from working air quality monitoring stations, satellite observations, and IoT devices can turn the predictions into dynamic responsive ones. Real-time data integration can enable the updating of air quality forecasts in real-time, which might prove critical in decision making and well-timed public health interventions. Technologies that can help in the continuous ingestion of data and model updating, so that predictions reflect the current environmental situation and new, emerging sources of pollution, include streaming data platforms and real-time analytics.

**2.Advanced Machine Learning Techniques**

With the advancements in machine learning technologies, the future development of air quality prediction should consider further advanced techniques for model performance improvement. While traditional algorithms like decision trees and random forests are effective, more sophisticated methods, such as deep learning and ensemble models, show much potential. Deep learning models, particularly the Recurrent Neural Networks and Long Short-Term Memory networks, capture the temporal dependencies and complex patterns in time-series data that are crucial for air quality prediction. Moreover, hybrid models that combine different machine learning techniques or tap exogenous data sources enhance predictive accuracy and robustness. Research in new algorithms and applying them to air quality prediction may be done for more accurate and reliable forecast.

**3.Personalized Air Quality Forecast**

Another promising line of future development involves developing personalized air quality forecasts. Using personal health information, location-specific information, and historical exposure patterns, predictive models generate personal recommendations and alerts for a person, by his health condition and environmental exposures. For instance, customized forecasts could be given to people who have respiratory or allergic conditions, with a detail of the risks involved and precautionary measures. By fusing health data with models of air quality, more individual effects of pollution are understood, and it can, at the same time, provide interventions tailored towards the prevention of risk to health.

**4.Provide more diverse sources and features of data**

Additional sources of data and features can also be added to future projects on air quality prediction to make it more comprehensive and accurate. This can be achieved by adding data on traffic patterns, industrial activities, and meteorological forecasts in order to get an all-rounded view on pollution sources and impacts. For example, traffic congestion and vehicle emission data would allow for understanding of the locations and times of heavy pollution, while weather forecasts would modulate factors such as wind direction and precipitation, which affect pollutant dispersion. Further, the incorporation of satellite data with remote sensing technologies would offer insight into regional and global trends in pollution, hence making the model's predictions more accurate. Additional sources of data and features can make air quality forecasting more accurate, and thus more context-aware.

**5.Policy and Regulation Support**

Predicting models of air quality can contribute enormously to the goals in support of environmental policy and regulation. Predictive models can point out regions of high pollution risk and simulate the effect of proposed regulations, making valuable contributions to insights for policy people. For instance, models can simulate scenarios of emission reduction or assess the impact of newly introduced environmental policies on air quality. Air quality prediction models can provide evidence-based policy recommendations and scenario analyses as tools for the formulation and review of effective policies and regulations for improvement in air quality and public health.

**6.Increasing public awareness and participation**

The future for the enhancement in scope for the prediction of air quality would also bring improvement in awareness and involvement of the public through the use of user-friendly tools. Mobile applications and online platforms can be developed to inform and alert, providing real-time air quality information, forecasts, and health recommendations for empowering users to make informed decisions in daily life regarding exposure to air pollution. Awareness programs for the public through educational campaigns andoutreach could also be included in the public engagement initiative. Given the democratization of accessibility and usability of information on air quality, individuals and communities can make informed decisions to reduce personal exposure and drive further initiatives toward air quality improvement.

**7.Academic Collaboration**

Research institutions and institutions involved in academics can team up in driving innovation and improvements in the prediction of air quality. Collaboration betweensome of these universities or research institutions would be very important in coming up with new algorithms, evolving state-of-the-art technologies, and discovering new sources of data. The collaborative research projects can also be used in validating and refining prediction models through tests and evaluations. By working in active collaboration with higher-level research facilities, air quality prediction projects can always be at the forefront of available technologies and support the general study of environmental science and public health.

1. **Scalability and Adaptability**

For their broader applicability, it is necessary that models of air quality prediction be scaled up or down and adaptable to varying regional contexts. Models developed for one geographic area cannot be directly applied to others, due to variations in sources of pollution, meteorological conditions, and regulatory environments. This calls for future development that considers adaptable modeling approaches to different locations and scales—from local neighborhoods to global regions. Finally, scalability considerations can also involve model optimizations regarding performance and resource efficiency toward large datasets and real-time processing demands. Such can make the air quality prediction system more deployable and effective in various contexts.

**9.Integrating with Health Impact Models**

Finally, integration between air quality prediction models and health impact models can be used to build a holistic picture of the impacts of pollution on public health. Health impact models use air quality data to provide estimates of the health outcome and related economic costs that might be associated with exposure to pollutants. Coupling these models with air quality predictions, one could provide estimates of potential health risks and benefits of alternative pollution control strategies and interventions. Integration at this level would enhance public health planning and resource allocation by providing actionable insight into how changes in air quality might impact population health.

58

**CHAPTER 7**

**REFERENCES**

1. *Méndez, Manuel, Mercedes G.Merayo , and Manuel Núñez. "Machine learning algorithms to forecast air quality: a survey." Artificial Intelligence Review 56, no. 9 (2023): 10031-10066.*
2. *Ravindiran , Gokulan , Gasim Hayder, Karthick Kanagarathinam, Avinash Alagumalai, and Christian Sonne. "Air quality prediction by machine learning models: A predictive study on the indian coastal city of Visakhapatnam." Chemosphere 338 (2023): 139518.*
3. *Gupta, N. Srinivasa, Yashvi Mohta, Khyati Heda, Raahil Armaan, B. Valarmathi, and G. Arulkumaran. "Prediction of air quality index using machine learning techniques: a comparative analysis." Journal of Environmental and Public Health 2023, no. 1 (2023): 4916267.*
4. *Kumar, K., and B. P. Pande. "Air pollution prediction with machine learning: a case study of Indian cities." International Journal of Environmental Science and Technology 20, no. 5 (2023): 5333-5348.*
5. *Chen, Ming, Pengcheng Xu, Zepeng Liu, Fang Liu, Haiqiu Zhang, and Shoulei Miao. "Air pollution prediction based on optimized deep learning neural networks: PSO-LSTM." Atmospheric Pollution Research 16, no. 3 (2025): 102413.*
6. *M. S. Ram, C. Reshmasri, S. Shahila and J. V. P. Saketh, "Air Quality Prediction using Machine Learning Algorithm," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 316-321, doi: 10.1109/ICSCDS56580.2023.10105063*
7. *Liao, Haibin, Li Yuan, Mou Wu, and Hongsheng Chen. "Air quality prediction by integrating mechanism model and machine learning model." Science of The Total Environment 899 (2023): 165646.*
8. *Natarajan, Suresh Kumar, Prakash Shanmurthy, Daniel Arockiam, Balamurugan Balusamy, and Shitharth Selvarajan. "Optimized machine learning model for air quality index prediction in major cities in India." Scientific Reports 14, no. 1 (2024): 6795.*
9. *Kang, Gaganjot Kaur, Jerry Zeyu Gao, Sen Chiao, Shengqiang Lu, and Gang Xie. "Air quality prediction: Big data and machine learning approaches." Int. J. Environ. Sci. Dev 9, no. 1 (2018): 8-16.*
10. *Pak, Abbas, Abdullah Kaviani Rad, Mohammad Javad Nematollahi, and Mohammadreza Mahmoudi. "Application of the Lasso regularisation technique in mitigating overfitting in air quality prediction models." Scientific Reports 15, no. 1 (2025): 547.*
11. *Jayaraman, Shrikar, and Abirami S. "Enhancing urban air quality prediction using time-based-spatial forecasting framework." Scientific Reports 15, no. 1 (2025): 4139.*
12. *Yu, Chengqing, Fei Wang, Yilun Wang, Zezhi Shao, Tao Sun, Di Yao, and Yongjun Xu. "MGSFformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction." Information Fusion 113 (2025): 102607.*
13. *Pak, Abbas, Abdullah Kaviani Rad, Mohammad Javad Nematollahi, and Mohammadreza Mahmoudi. "Application of the Lasso regularisation technique in mitigating overfitting in air quality prediction models." Scientific Reports 15, no. 1 (2025): 547.*
14. *Zhang, Chengxin, Xinhan Niu, Hongyu Wu, Zhipeng Ding, Ka Lok Chan, Jhoon Kim, Thomas Wagner, and Cheng Liu. "Unleashing the potential of geostationary satellite observations in air quality forecasting through artificial intelligence techniques." Atmospheric Chemistry and Physics 25, no. 2 (2025): 759-770.*
15. *Zhong, Hui, Di Chen, Pengqin Wang, Wenrui Wang, Shaojie Shen, Yonghong Liu, and Meixin Zhu. "Predicting On-Road Air Pollution Coupling Street View Images and Machine Learning: A Quantitative Analysis of the Optimal Strategy." Environmental Science & Technology (2025).*
16. *Iskandaryan, Ditsuhi, Francisco Ramos, and Sergio Trilles. "Air quality prediction in smart cities using machine learning technologies based on sensor data: a review." Applied Sciences 10, no. 7 (2020): 2401.*
17. *Hardini, Marviola, Mochamad Heru Riza Chakim, Lena Magdalena, Hiroshi Kenta, Ageng Setiani Rafika, and Dwi Julianingsih. "Image-based air quality prediction using convolutional neural networks and machine learning." Aptisi Transactions on Technopreneurship (ATT) 5, no. 1Sp (2023): 109-123.*
18. *Castelli, Mauro, Fabiana Martins Clemente, Aleš Popovič, Sara Silva, and Leonardo Vanneschi. "A machine learning approach to predict air quality in California." Complexity 2020, no. 1 (2020): 8049504.*
19. *Peng, Huiping. "Air quality prediction by machine learning methods." PhD diss., University of British Columbia, 2015.*

1. *Gladkova, Ekaterina, and Liliya Saychenko. "Applying machine learning techniques in air quality prediction." Transportation Research Procedia 63 (2022): 1999-2006.*

**PUBLICATION CERTIFICATE**