International Journal for Modern Trends in Science and Technology
Volume 12, Issue 06, pages 261-273.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue06.html
DOI: https://doi.org/10.5281/zenodo.20739547

Deep Learning Based Real-Time Air Quality Prediction and Pollution Monitoring System

M. J. V. B. Suma, Nallamothu Divya Durga, Thanniru Sivaramakrishna, Cheerla Nanda Kishore

Department of Information Technology Swarnandhra College of Engineering & Technology, Seetharamapuram, Narsapur, Andhra Pradesh, INDIA.

To Cite this Article
M. J. V. B. Suma, Nallamothu Divya Durga, Thanniru Sivaramakrishna & Cheerla Nanda Kishore (2026). Deep Learning Based Real-Time Air Quality Prediction and Pollution Monitoring System. International Journal for Modern Trends in Science and Technology, 12(06), 261-273. https://doi.org/10.5281/zenodo.20739547

Article Info
Received: 16 May 2026; Revised: 07 May 2026; Accepted: 12 June 2026.

KEYWORDS

Air Quality, AQI, Air Pollution, Machine Learning, XGBoost, Real-Time Monitoring, Prediction System.

ABSTRACT

Air pollution is a serious problem that affects human health and the environment. It is mainly caused by vehicles, industries, and other human activities. Harmful gases like PM2.5, PM10, CO, NO_2, SO_2, and O_3 increase pollution levels and make the air unsafe to breathe. Existing air quality systems only show current pollution levels and do not predict future conditions.This project develops a Real-Time Air Quality Prediction and Pollution Monitoring System using machine learning techniques. The system uses XGBoost to predict the Air Quality Index (AQI) accurately. It also uses TF-IDF for feature processing and cosine similarity to compare current data with past data for better results. The system collects data from sensors, datasets, and user inputs. The data is cleaned and processed before giving it to the model. The model then predicts the AQI and classifies it into categories like Good, Moderate, and Unhealthy.The system also shows results in a simple way using graphs and provides alerts when pollution levels are high. A user-friendly interface is created using Streamlit for easy use.
This system helps people understand air quality, take precautions, and reduce pollution risks.

1. INTRODUCTION

Air pollution has emerged as one of the most serious environmental and public health challenges in recent decades. Rapid industrialization, urbanization, population growth, and increased vehicular emissions have significantly contributed to the deterioration of air quality across the globe. The presence of harmful pollutants such as particulate matter (PM2.5 and PM10), carbon monoxide(CO),nitrogen dioxide (NO_2),sulfur dioxide (SO_2), and ozone(O_3) has led to severe health complications, including respiratory diseases, cardiovascular disorders, and even premature deaths.

Traditional air quality monitoring systems primarily focus on measuring pollutant concentrations using fixed monitoring stations. While these systems provide valuable real-time data, they lack the capability to predict future air quality conditions. This limitation restricts proactive decision-making, making it difficult for authorities and individuals to take preventive measures in advance. With the advancement of Artificial Intelligence (AI) and Machine Learning (ML), there has been a paradigm shift in environmental monitoring systems.

Air quality These technologies enable the analysis of large-scale environmental datasets, identification of hidden patterns, and prediction of future pollution levels with high accuracy. In particular, machine learning algorithms such as XG Boost have proven to be highly effective in handling complex, non-linear environmental data.

This project introduces a Deep Learning Based Real-Time Air Quality Prediction and Pollution Monitoring System, which integrates machine learning techniques, data preprocessing methods, and similarity-based analysis to provide accurate predictions of. The system utilizes TF-IDF (Term Frequency-Inverse Document Frequency) for feature extraction, XG Boost for classification and prediction, and cosine similarity for improving the relevance of future predictive analytics, the system aims to support environmental management, public awareness, and policy-making. Air pollution is a global issue affecting both developed and developing countries.According to environmental studies, millions of people worldwide are exposed to unsafe air quality levels every day. Major cities experience high pollution due to industrial emissions, transportation, and urban activities. Countries with rapid industrial growth face severe pollution problems. Urban areas especially suffer from high concentrations of particulate matter and harmful gases. Climate change has also contribute to the worsening of air quality by increasing temperature and affecting atmospheric conditions. Air pollution is not limited to outdoor environments; indoor air pollution is also a major concern due to poor ventilation and the use of harmful substances. Therefore, continuous monitoring and prediction systems are essential to manage and reduce pollution levels globally.

1.1 EVOLUTION OF AIR QUALITY MONITORING SYSTEM

Air quality monitoring systems have developed significantly over time, moving from simple manual methods to advanced intelligent systems. This evolution reflects the increasing need for accurate, real-time, and predictive analysis of environmental conditions.

In the early stage, air quality monitoring was performed using manual methods. Air samples were collected using basic instruments and analyzed in laboratories. This process was time-consuming and required significant human effort. It also lacked real-time capabilities, as results were available only after laboratory analysis. Due to these limitations, manual monitoring methods were not suitable for continuous or large-scale environmental monitoring. With technological advancements, automated monitoring systems were introduced.

These systems used electronic sensors to measure pollutant levels continuously. Automated systems improved accuracy and reduced human effort by providing real-time data collection and storage. However, these systems were expensive to install and maintain, and they were usually limited to specific locations. Most importantly, they only provided current pollution levels and did not have the ability to predict future air quality.

The introduction of sensor-based and Internet of Things (IoT) technologies further improved air quality monitoring. IoT-based systems use smart sensors that can collect and transmit data wirelessly. These systems allow remote monitoring and provide continuous real-time data from multiple locations. They are more flexible and cost-effective compared to traditional monitoring stations. However, issues such as sensor calibration, data noise, and maintenance challenges still exist.

In recent years, the integration of cloud computing and big data analytics has enhanced the capabilities of air quality monitoring systems. Large volumes of environmental data collected from sensors can now be stored and processed efficiently using cloud platforms. This enables faster data analysis, improved accessibility, and better decision-making. Data visualization tools also help in presenting complex data in a user-friendly manner through graphs, dashboards, and alerts.

Furthermore, the use of machine learning and artificial intelligence has transformed air quality monitoring from a reactive system to a predictive system. Advanced algorithms can analyze historical and real-time data to forecast future pollution levels with high accuracy. These intelligent systems can identify patterns, detect anomalies, and provide early warnings for hazardous air conditions.

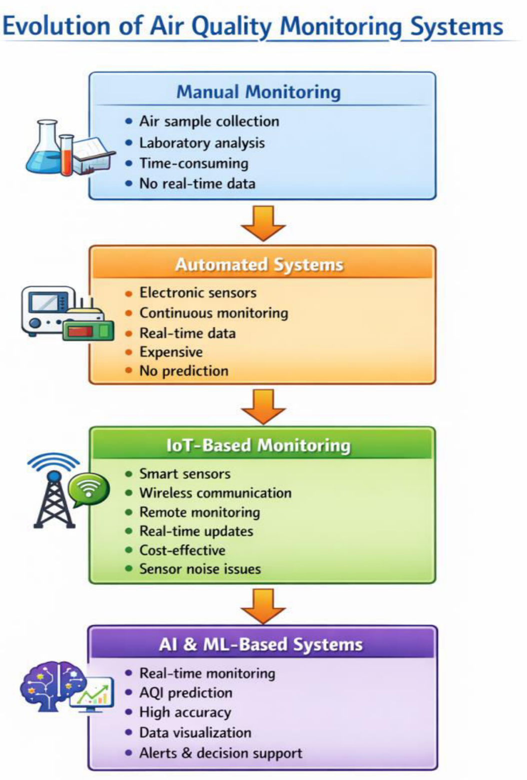

Fig 1.1: Evolution of Air Quality Monitoring Systems

1.2 MACHINE LEARNING IN AIR QUALITY PREDICTION

Machine learning is an important technology used in air quality prediction systems. It helps in analyzing large amounts of environmental data and predicting future pollution levels. Unlike traditional methods, machine learning algorithms can learn from past data and identify patterns without manual programming. In air quality prediction, data such as pollutant levels (PM2.5, PM10, CO, NO_2, SO_2, O_3), weather conditions, and time-based information are used as input.

The machine learning model processes this data and learns the relationship between different factors. Based on this learning, it predicts the Air Quality Index (AQI) accurately. Different machine learning algorithms are used for prediction, such as decision trees, random forests, and XGBoost. Among these, XGBoost is widely used because it provides high accuracy, handles missing data, and works efficiently with large datasets. Machine learning models can also handle complex and non-linear relationships in the data, which improves prediction performance. Another advantage of machine learning is its ability to provide real-time predictions. It can analyze incoming data continuously and update predictions instantly. This helps in giving early warnings about dangerous pollution levels and allows people to take preventive measures. Machine learning also uses techniques like feature extraction and similarity analysis to improve results. These techniques help selecting important data and comparing current conditions with past patterns. Overall, machine learning provides a powerful and efficient method for air quality prediction.

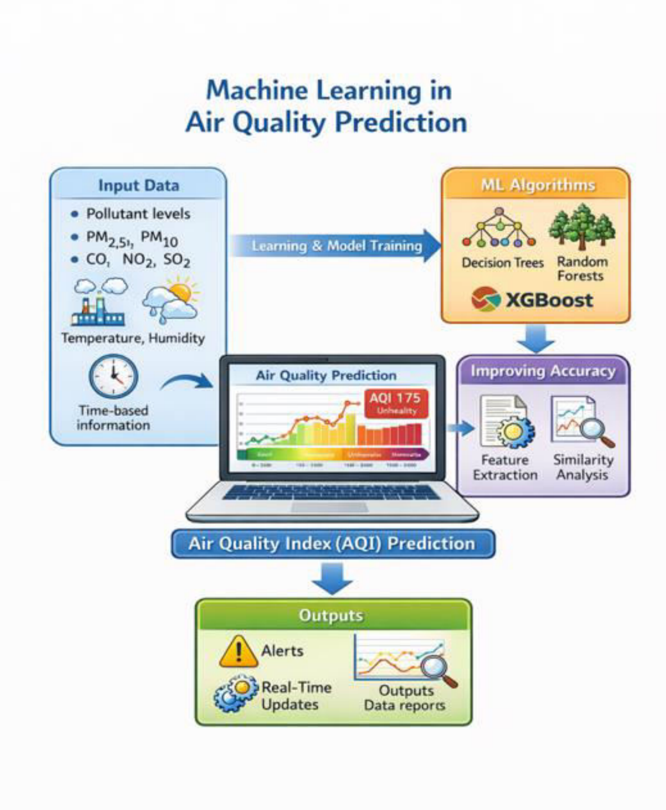

Fig 1.2: Machine Learning in Air Quality Prediction

1.3 ROLE OF DEEP LEARNING IN ENVIRONMENTAL MONITRING

Deep learning plays a significant role in environmental monitoring by providing advanced methods for analyzing complex and large-scale environmental data. It is a subset of machine learning that uses artificial neural networks with multiple layers to learn patterns and relationships in data. Unlike traditional machine learning techniques, deep learning can automatically extract important features from raw data, which improves accuracy and efficiency.

In environmental monitoring, deep learning is used to analyze data related to air quality, weather conditions, pollution levels, and climate changes. It can process large volumes of data collected from sensors, satellites, and monitoring stations. Deep learning models such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) are commonly used for this purpose.

One of the major advantages of deep learning is its ability to handle complex and non-linear relationships between different environmental factors. For example, pollution levels are influenced by multiple variables such as temperature, humidity, wind speed, and human activities. Deep learning models can learn these relationships and provide accurate predictions of air quality. Deep learning is also useful for time-series forecasting, where it predicts future pollution levels based on past data. Models like Long Short-Term Memory (LSTM) networks are especially effective in capturing temporal patterns and trends in environmental data.

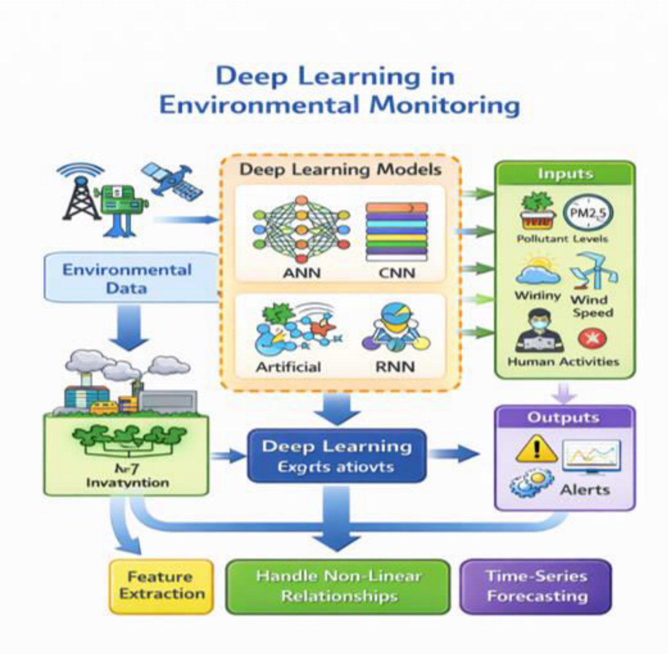

Fig 1.3: Deep Learning in Environmental Monitoring.

1.4 REAL-TIME AIR QUALITY MONITORING

Real-time air quality monitoring is the process of continuously measuring and analyzing air pollution levels as they occur. It provides immediate information about the current condition of the air, which helps in understanding pollution levels at any given moment. This type of monitoring is essential in urban and industrial areas where pollution levels can change rapidly due to traffic, weather conditions, and human activities.

In real-time monitoring systems, data is collected using sensors that detect pollutants such as PM2.5, PM10, CO, NO_2, SO_2, and O_3. These sensors are connected to monitoring devices or IoT systems that transmit data instantly to a central system. The collected data is then processed and displayed to users through dashboards, graphs, or mobile applications.

One of the main advantages of real-time monitoring is that it provides instant updates about air quality. This helps individuals and authorities to take immediate actions when pollution levels become dangerous. For example, people can avoid outdoor activities, and government agencies can issue alerts or implement control measures.

Real-time monitoring systems also support data analysis by continuously collecting large amounts of data. This data can be used for identifying trends, understanding pollution sources, and improving prediction models. When combined with machine learning techniques, real-time data can be used to forecast future air quality levels.

2. LITERATURE SURVEY

Air pollution prediction and monitoring have gained significant attention in recent years due to increasing environmental concerns and health risks. Researchers across the globe have explored various techniques ranging from traditional statistical models to advanced machine learning and deep learning approaches for predicting air quality. This chapter reviews existing research works related to air quality prediction, machine learning applications, deep learning techniques, and real-time monitoring systems. The primary objective of this review is to analyze the strengths and limitations of existing methods and to justify the need for an improved system that integrates machine learning techniques such as XGBoost, TF-IDF, and cosine similarity for accurate and real-time air quality prediction.

2.1 AIR QUALITY REDICTION USING TRADITIONAL METHODS

Traditional methods for air quality prediction mainly rely on statistical techniques and mathematical models to estimate pollution levels. These methods use historical data and predefined formulas to analyze and predict the Air Quality Index (AQI). Before the development of machine learning, these approaches were widely used in environmental monitoring systems.

One of the commonly used traditional methods is linear regression, which establishes a relationship between pollutant levels and environmental factors. It predicts air quality based on past trends. Another method is time-series analysis, such as ARIMA (Auto Regressive Integrated Moving Average), which is used to forecast future pollution levels based on previous data patterns.

These methods are simple, easy to implement, and require less computational power. They are useful when the data is small and follows a linear pattern. Traditional models are also helpful in understanding basic relationships between variables.

However, traditional methods have several limitations. They are not effective in handling complex and non-linear relationships between different environmental factors. Air pollution is influenced by many variables such as weather conditions, traffic, and industrial activities, which cannot be accurately captured using simple statistical models.

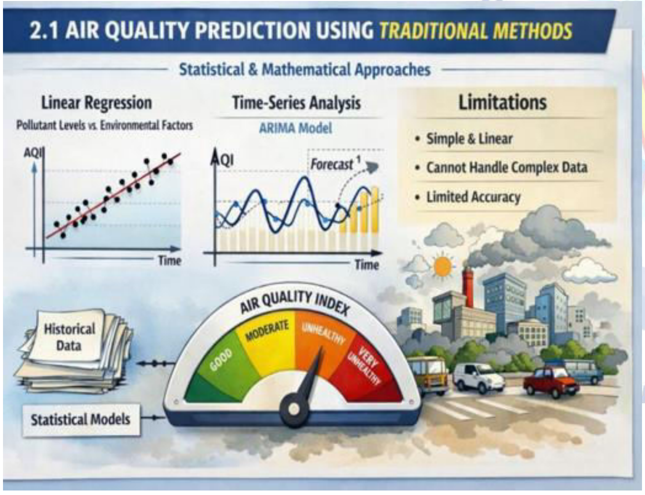

Fig 2.1: Air Quality Prediction Using Traditional Method

2.2 MACHINE LEARNING APPROACH FOR AIR QUALITY PREDICTION

With the advancement of computational power, machine learning techniques have been widely adopted for air quality prediction.

Supervised Learning Models

Several studies have used supervised learning algorithms such as:

- Support Vector Machines (SVM)
- Decision Trees
- Random Forest
- Gradient Boosting

These models use labeled datasets to predict AQI based on input features like pollutant concentrations and weather conditions.

Advantages

- Better accuracy compared to traditional models
- Ability to handle large datasets
- Improved generalization

Limitations

- Require large labeled datasets
- Sensitive to data quality
- Computational complexity

Among these, ensemble learning methods like Gradient Boosting and XGBoost have shown superior performance due to their ability to combine multiple weak learners into a strong predictive model.

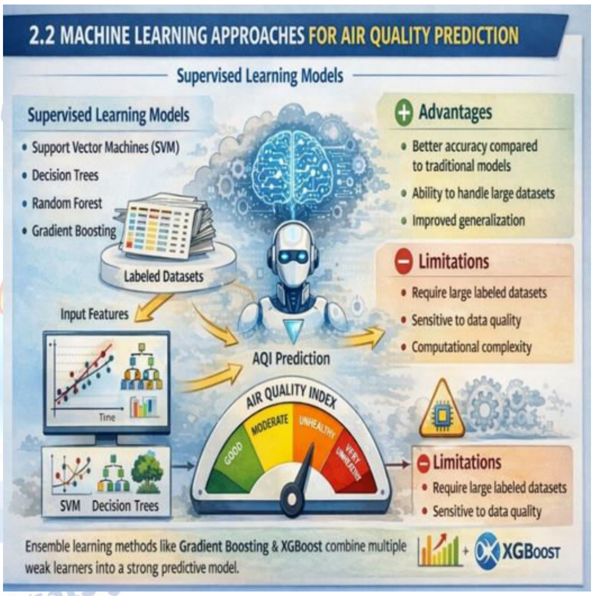

Fig 2.2: Machine Learning Approaches for Air Quality Prediction

2.3 XGBOOST IN ENVIRONMENTAL DATA ANALYSIS

XGBoost (Extreme Gradient Boosting) is one of the most powerful machine learning algorithms used in environmental data analysis.

Key Features

- High prediction accuracy
- Regularization to prevent overfitting
- Ability to handle missing data
- Fast computation using parallel processing

Several research studies have demonstrated the effectiveness of XGBoost in predicting air quality levels. It has outperformed traditional algorithms in terms of accuracy and efficiency.

Applications in Air Quality Prediction

- AQI forecasting
- Pollution level classification
- Environmental risk assessment

Due to these advantages, XGBoost is selected as the core prediction model in the proposed system.

2.4 DEEP LEARNING TECHNIQUES FOR AIR QUALITY PREDICTION

Deep learning models have gained popularity due to their ability to learn complex patterns from large datasets.

Common Deep Learning Models

- Artificial Neural Networks (ANN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Long Short-Term Memory (LSTM)

Applications

- Time-series forecasting of pollution levels
- Feature extraction from environmental data
- Real-time prediction systems

Advantages

- High accuracy
- Ability to model complex relationships
- Suitable for large-scale data

Challenges

- Requires high computational resources
- Complex model training
- Risk of overfitting

Although deep learning provides high accuracy, it may not always be suitable for real-time systems with limited computational resources. Therefore, this project adopts a hybrid approach using efficient machine learning techniques.

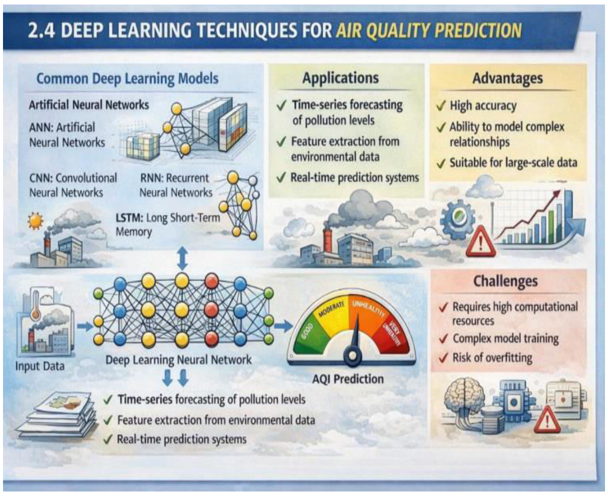

Fig 2.4: Deep Learning For Air Quality Prediction

2.5 REAL-TIME AIR QUALITY MONITORING SYSTEM

Real-time monitoring systems play a crucial role in tracking environmental conditions.

Existing Systems

- IoT-based sensor networks
- Government monitoring stations
- Cloud-based data platforms

Features

- Continuous data collection
- Instant AQI calculation
- Data visualization

Limitations

- Lack of prediction capability
- High infrastructure cost
- Limited scalability

To overcome these limitations, modern systems integrate machine learning models to provide predictive insights along with real-time monitoring.

3. PROPOSED METHODOLOGY

The proposed system addresses limitations of existing monitoring tools by enabling real-time AQI prediction alongside current pollution tracking. Unlike traditional systems that primarily provide descriptive statistics of present air conditions, this solution integrates predictive analytics to forecast upcoming air quality levels. By combining live sensor feeds with historical datasets, the system ensures both immediacy and contextual depth in analysis. This dual capability empowers users not only to understand the current pollution scenario but also to anticipate future changes and take preventive measures in advance.

Data is collected from multiple sources, including IoT-based environmental sensors, publicly available air quality datasets, and optional user inputs. Key parameters such as PM2.5, PM10, CO, NO_2, SO_2, O_3, temperature, and humidity are continuously monitored. To ensure reliability, the raw data undergoes preprocessing steps such as noise removal, missing value imputation, normalization, and outlier detection. This data cleaning phase improves the overall quality of inputs, ensuring that subsequent modeling and forecasting processes are accurate and robust.

For feature engineering, the system applies TF-IDF (Term Frequency–Inverse Document Frequency) to transform pollutant patterns and contextual environmental data into structured numerical representations. Although TF-IDF is traditionally used in text processing, here it is adapted to quantify the significance of pollutant trends across time windows. Cosine similarity is then employed to compare current environmental conditions with historical pollution patterns. By identifying closely matching historical scenarios, the system enhances prediction reliability through pattern-based contextual referencing.

The refined features are fed into an XGBoost-based machine learning model for AQI forecasting. XGBoost is selected for its high efficiency, scalability, and strong performance in handling structured environmental datasets. The model is trained on historical air quality records and optimized using cross-validation techniques to minimize prediction error. Based on predicted AQI values, the system categorizes air quality into standard classifications such as Good, Moderate, and Unhealthy, making the results easy to interpret for non-technical users.

To ensure usability and accessibility, the system incorporates a user-friendly Streamlit interface. The dashboard presents interactive visualizations, including time-series graphs, pollutant concentration trends, comparative historical analysis, and real-time AQI indicators. Color-coded alerts and notifications are triggered when pollution levels exceed safe thresholds. These alerts provide actionable recommendations, such as limiting outdoor activities or wearing protective masks, thereby supporting proactive health decisions.

Overall, the proposed framework combines real-time monitoring, intelligent forecasting, and intuitive visualization into a single integrated platform. By leveraging machine learning and similarity-based analysis, it enhances predictive accuracy while maintaining interpretability. The system is scalable and can be expanded to include additional environmental factors or integrated with smart city infrastructure. In doing so, it contributes to more informed public health management, environmental awareness, and data-driven decision-making for sustainable urban living.

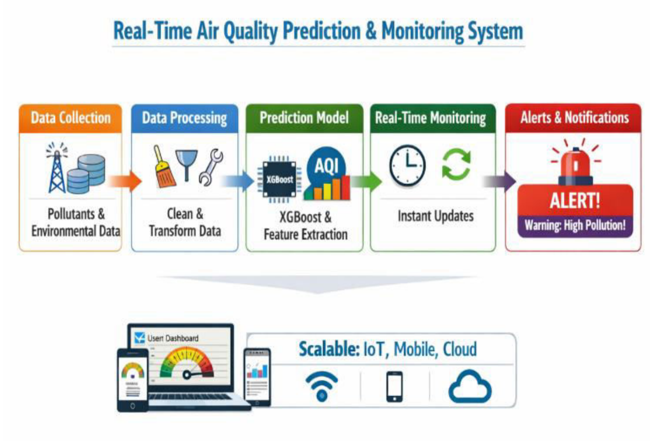

Fig 3.1: Real-Time Air Quality Prediction & Monitoring System

3.1 SYSTEM ARCHITECTURE

The modular architecture of the proposed system is designed to ensure scalability, flexibility, and efficient real-time performance. The workflow begins with the data collection layer, where information is gathered from multiple sources such as IoT-based environmental sensors, publicly available air quality datasets, and optional user-provided inputs. These inputs include pollutant concentrations (PM2.5, PM10, CO, NO_2, SO_2, O_3) along with meteorological parameters like temperature and humidity. By integrating diverse data streams, the system builds a comprehensive environmental profile that supports accurate monitoring and forecasting.

Once collected, the data moves to the preprocessing layer, which plays a crucial role in enhancing data quality and reliability. This stage involves cleaning operations such as handling missing values, removing noise, detecting and treating outliers, and ensuring consistency across datasets. Normalization techniques are applied to scale the data into a uniform range, preventing bias during model training. The preprocessing module operates automatically and continuously, enabling smooth handling of large, real-time data streams without compromising accuracy.

After preprocessing, the system proceeds to the feature extraction phase, where TF-IDF is applied to transform pollutant trend information into structured numerical representations. By analyzing pollutant frequency patterns across defined time intervals, TF-IDF assigns importance weights to significant environmental indicators. This approach allows the system to capture both dominant and emerging pollution trends effectively. The extracted features provide a compact yet informative representation of the environmental data, which strengthens the predictive capability of the subsequent model.

The processed features are then fed into the XGBoost prediction model. This module is responsible for forecasting AQI values based on learned relationships within historical data. XGBoost's gradient boosting framework enhances prediction accuracy by iteratively minimizing errors and optimizing performance. Its ability to handle nonlinear relationships and large datasets makes it well-suited for environmental modeling. The modular nature of this component allows for easy retraining or replacement with alternative algorithms if future improvements are required.

Finally, the output layer presents the predicted AQI values and pollution insights through an interactive Streamlit interface. Visual elements such as real-time graphs, trend charts, and categorized AQI levels (e.g., Good, Moderate, Unhealthy) enable users to quickly interpret results. Automated alert mechanisms notify users when pollution levels exceed safe thresholds, supporting proactive decision-making. As illustrated in, this end-to-end pipeline emphasizes continuous data flow and real-time processing, ensuring that the system remains scalable and responsive to evolving environmental conditions.

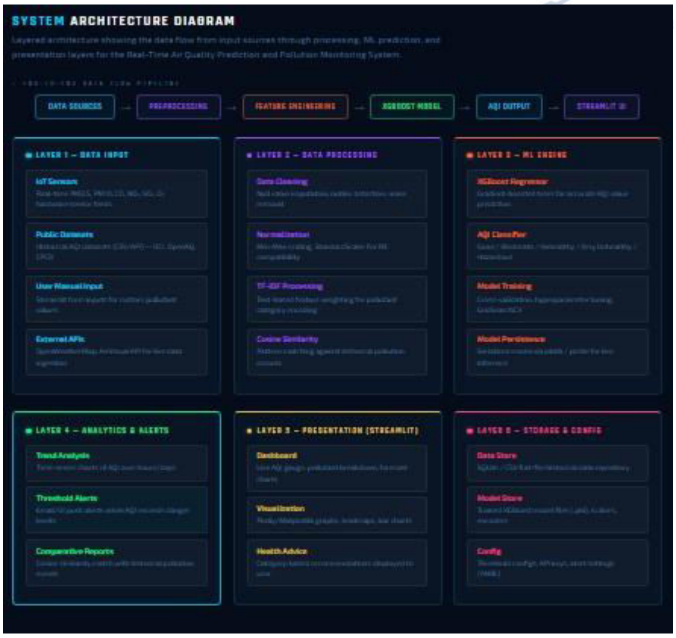

Fig 3.1 Proposed System Architecture

The System Architecture Diagram presents the layered structural design of the Real-Time Air Quality Prediction and Pollution Monitoring System. It illustrates how data flows from multiple input sources through processing, machine learning prediction, and final visualization.

3.1.1 Data Input Layer

The Data Input Layer is responsible for collecting raw environmental data from diverse sources. These include:

- IoT-based air quality sensors providing real-time pollutant readings (PM2.5, PM10, CO, NO_2, SO_2, O_3, temperature, and humidity).
- Public air quality datasets for historical analysis.
- User-provided manual inputs for custom AQI prediction.
- External APIs for supplementary environmental data.

This multi-source integration ensures both real-time responsiveness and historical contextual analysis.

3.1.2 Data Preprocessing Layer

The Data Preprocessing Layer enhances data quality before model training and prediction. The following operations are performed:

- Data cleaning to remove missing values, duplicates, and noise.
- Normalization to scale numerical values for consistent model input.
- TF-IDF feature extraction to determine weighted importance of pollutant trends.
- Cosine similarity computation to compare current environmental conditions with historical pollution patterns.

This layer ensures accurate, reliable, and structured data for the machine learning engine.

3.1.3 Machine Learning Layer

The Machine Learning Layer forms the analytical core of the system. It consists of:

- XGBoost regression model for AQI value prediction.
- AQI classification module to categorize predicted values into standard levels (Good, Moderate, Unhealthy, etc.).
- Model evaluation and optimization processes.
- Model persistence for saving trained models for future use.

This layer transforms processed environmental data into meaningful forecasts.

3.1.4 Analytics and Alert Layer

After AQI prediction and classification, the system:

- Performs trend analysis.
- Generates threshold-based pollution alerts.
- Provides health recommendations.
- Produces comparative reports.

This ensures proactive environmental awareness and timely decision-making.

3.1.5 Presentation Layer

The Presentation Layer is implemented using Streamlit. It provides:

- Interactive dashboards.
- Real-time AQI display.
- Graphical visualizations and trend charts.
- Alert notifications and advisory messages.

The layered architecture supports modularity, scalability, and maintainability.

3.2 USE CASE DIAGRAM

The document does not explicitly include a dedicated Use Case Diagram; however, the system's functional behavior is effectively represented through the Activity Diagram and the Sequence Diagram. These diagrams collectively illustrate user interactions and internal system operations, covering processes such as data input, preprocessing, AQI prediction, classification, and alert display. Although they differ in structure from a traditional use case diagram, they convey similar functional insights regarding how users interact with the system and how the system responds.

The Activity Diagram represents the overall workflow of the system in a step-by-step manner. It visually describes the sequential flow beginning with data collection from sensors or manual user input, followed by preprocessing steps such as cleaning and normalization. The process then moves to feature extraction, AQI prediction using the machine learning model, and classification into standard AQI categories. Decision nodes within the activity diagram indicate conditional checks, such as whether pollution levels exceed predefined thresholds, which determine if alerts or health recommendations should be triggered.

The Sequence Diagram, on the other hand, focuses on dynamic interactions between system components over time. It illustrates how the user interface communicates with backend modules such as the Data Collector, Preprocessor, Prediction Engine, and Alert Manager. The diagram highlights message exchanges, method invocations, and response flows that occur when a user requests AQI prediction or views dashboard analytics. This time-ordered representation clarifies the coordination among components and demonstrates how real-time processing is achieved.

Together, the activity and sequence diagrams serve as functional equivalents to a use case diagram by describing both the procedural workflow and the interaction logic of the system. While a use case diagram typically emphasizes actors and their associated functionalities, these diagrams provide a more detailed view of internal processing and control flow. They not only capture user-triggered operations but also depict automated system responses such as threshold checking, similarity comparison, and alert generation.Therefore, even in the absence of a standalone use case diagram, the documentation effectively communicates system functionality through these behavioral models. The activity diagram explains "what happens" during execution, while the sequence diagram explains "how it happens" through component interaction. Combined, they provide comprehensive insight into system operations, ensuring clarity in functional documentation and supporting a strong understanding of user-system interaction within the Real-Time Air Quality Prediction platform.

Fig 3.2 Use Case Diagram

The Use Case Diagram illustrates functional interactions between system actors and the Real-Time Air Quality Prediction Platform.

3.2.1Actors

End User

- Views real-time AQI dashboard.
- Enters pollutant values manually.
- Receives pollution alerts.
- Views health recommendations.
- Compares historical pollution data.
- Visualizes graphical trends.

Researcher

- Analyzes historical pollution patterns.
- Uses cosine similarity comparison tools.

Government Agency

- Uploads and configures datasets.
- Manages sensor data feeds.
- Retrains or updates machine learning models.

Admin

- Manages system configuration.
- Oversees operational control.

Sensor API

- Supplies real-time environmental data.

ML Engine

- Performs AQI prediction and classification.

3.2.2Major Use Cases

- View Real-Time AQI Dashboard
- Enter Pollutant Values
- Predict AQI Using XGBoost
- Classify AQI Category
- Receive Pollution Alerts
- View Health Recommendations
- Compare Historical Data
- Visualize Trend Graphs and Charts
- Upload/Configure Dataset
- Manage Sensor Feed
- Retrain/Update ML Model

The use case diagram emphasizes role-based access, automation of prediction processes, and real-time monitoring capabilities.

3.3 CLASS DIAGRAM

The class diagram presents the static structural design of the system by illustrating the core classes, their attributes, methods, and the relationships between them. It provides a blueprint of how different components are organized within the Real-Time Air Quality Prediction System. Unlike dynamic diagrams that focus on workflow or interaction, the class diagram emphasizes the structural composition of the system and the responsibilities assigned to each class. This static representation ensures clarity in object-oriented design and supports modular implementation.

The User class represents the primary actor interacting with the system. It contains attributes such as userID, name, role, and inputData, along with methods like enterPollutantValues(), requestPrediction(), and viewResults(). This class acts as the entry point for user-driven operations, initiating the data flow process. By encapsulating user-related functionalities, the design maintains separation between user interaction logic and backend processing modules.

The DataHandler class is responsible for managing data acquisition and storage. It includes attributes such as sensorData, datasetPath, and apiEndpoint, and methods like fetchSensorData(), loadDataset(), and storeData(). This class ensures that environmental data from sensors, APIs, or manual inputs is properly collected and made available for further processing. Its relationship with the Preprocessor class reflects the sequential flow of data from collection to cleaning and transformation.

The Preprocessor and FeatureExtractor classes handle data preparation and feature engineering. The Preprocessor class contains methods such as cleanData(), normalizeData(), and handleMissingValues(), ensuring the raw data is structured and consistent. The FeatureExtractor class applies TF-IDF weighting and cosine similarity computations through methods like extractFeatures() and computeSimilarity(). These classes are directly associated with the ML Model class, ensuring that only processed and meaningful features are passed to the predictive engine.

The ML Model (XGBoost) and Predictor classes form the analytical core of the system. The ML Model class includes attributes such as trainedModel and modelParameters, along with methods like trainModel(), saveModel(), and loadModel(). The Predictor class utilizes the trained model to generate AQI predictions through predictAQI() and classifyAQI() methods. Finally, the Output Display class manages visualization and alert presentation, providing methods like displayDashboard(), showAlerts(), and generateReports(). The relationships among these classes illustrate a clear data flow from user input to final output, demonstrating strong cohesion within modules and logical dependency between components, thereby ensuring maintainability and scalability of the system design.

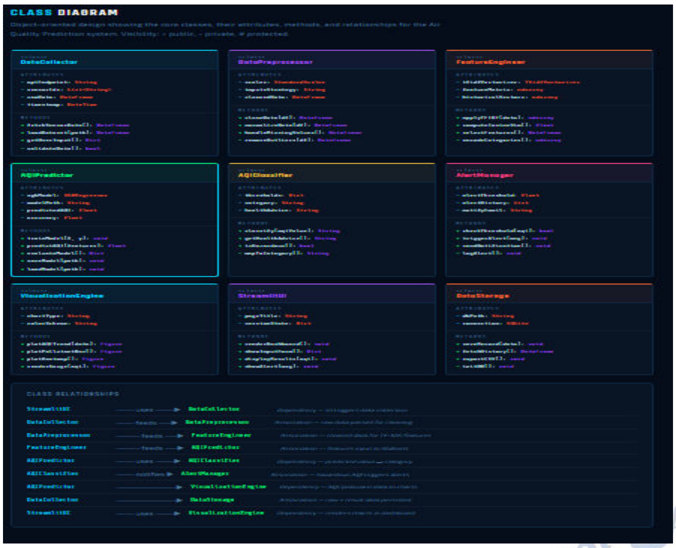

Fig 3.3 Class Diagram

The Class Diagram defines the internal object-oriented structure of the system. It illustrates classes, attributes, methods, and their relationships.

3.X.1 Core Classes

DataCollector

Responsible for acquiring environmental data.

Attributes: apiEndpoint, sensorData, dataset, userInput

Methods: fetchSensorData(), loadDataset(), getUserInput()

DataPreprocessor

Handles data cleaning and transformation.

Attributes: rawData, cleanedData, normalizedData

Methods: cleanData(), normalizeData(), handleMissingValues()

AQIPredictor

Implements the XGBoost regression model.

Attributes: xgModel, predictions, confidence

Methods: trainModel(), predictAQI(), evaluateModel()

AQIClassifier

Categorizes AQI values.

Attributes: thresholds, categories

Methods: classifyAQI(), getHealthAdvice()

AlertManager

Manages pollution notifications.

Attributes: alertThreshold, alertHistory

Methods: checkThreshold(), triggerAlert(), logAlert()

VisualizationEngine

Handles graphical output.

Attributes: chartType, dataSeries

Methods: plotTrend(), displayDashboard()

StreamlitUI

Provides user interface functionality.

Attributes: pageTitle, sessionState

Methods: renderDashboard(), displayAlerts()

DataStorage

Manages data persistence.

Attributes: dbPath, connection

Methods: saveSensorData(), saveModel(), loadData()

3.2.3Class Relationships

- DataCollector supplies data to DataPreprocessor.
- DataPreprocessor feeds structured data to AQIPredictor.
- AQIPredictor passes predicted AQI values to AQIClassifier.
- AQIClassifier interacts with AlertManager for notifications.
- VisualizationEngine displays outputs to StreamlitUI.
- DataStorage supports data saving and retrieval across modules.

The class diagram demonstrates modular design, high cohesion within classes, and low coupling between components, ensuring system scalability and maintainability.

4. RESULTS

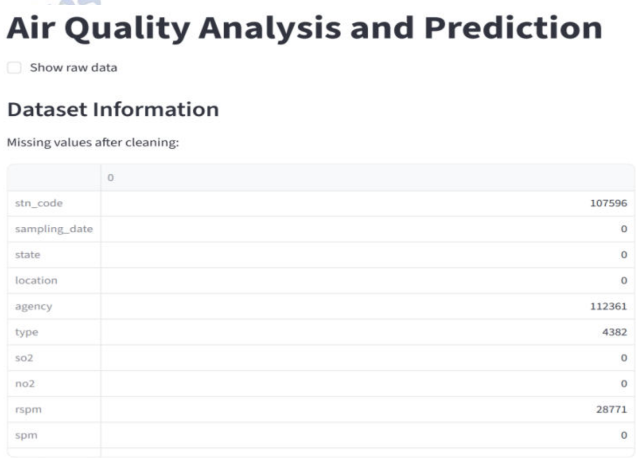

Fig 4.1: Air Quality Analysis and Prediction

Summary Statistics (Cleaned Data):

	so2	no2	rspm	spm	pm2_5	SOi	Noi	SPMi	AQI
count	170704	170704	141933	170704	0	170704	170704	170704	170704
mean	12.1005	28.1334	109.6698	226.8775	None	15.0499	35.0338	192.0023	192.3153
std	11.0324	18.9923	75.8361	153.5216	None	13.0467	22.9622	115.1798	114.8083
min	0	0	0	0	None	0	0	0	0
25%	5.5	15.9	58	115	None	6.875	19.875	110	110
50%	8.9	24	92	194	None	11.125	30	162.6667	162.6667
75%	15.1525	34.9	141	306	None	18.9406	43.625	256	256
max	273.3	484.3	6307.0333	3380	None	164.4333	470.25	1086.0465	1086.0465

Fig 4.2: Summary Statistics(Cleared Data)

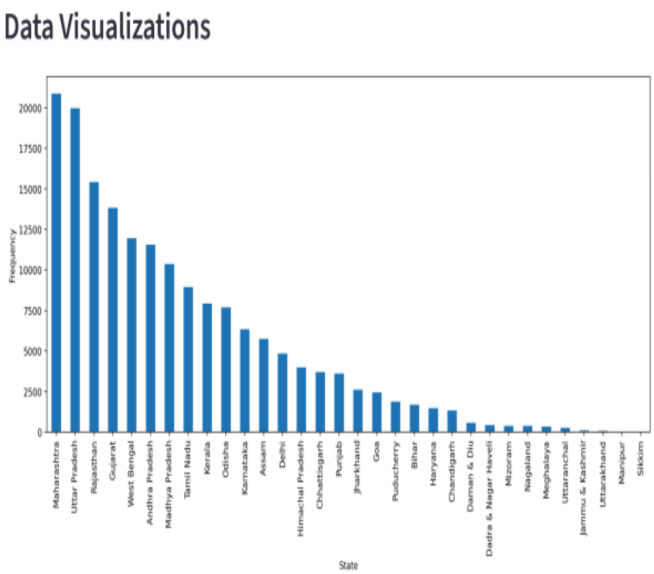

Fig 4.3: Data visualization

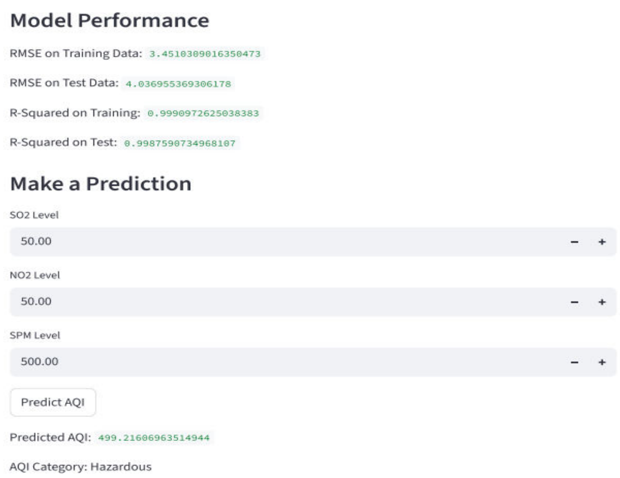

Fig 4.4: Model Performance

5. CONCLUSION

The project demonstrates the use of advanced technologies such as XGBoost, TF-IDF, and cosine similarity to analyze environmental data and predict the Air Quality Index (AQI). These techniques improve the accuracy and efficiency of the system compared to traditional methods.

The system is capable of collecting input data, processing it through multiple stages, and generating meaningful outputs in the form of AQI values and categories. The integration of a user-friendly interface using Streamlit makes the system easy to use for both technical and non-technical users.

One of the key strengths of the system is its ability to provide real-time monitoring and prediction, which helps users understand current pollution levels as well as future conditions. The alert mechanism further enhances the system by notifying users when pollution levels become unsafe, allowing them to take preventive measures.

The project also highlights the importance of data-driven decision-making in environmental monitoring. By analyzing historical and real-time data, the system helps in identifying pollution patterns and trends, which can be useful for researchers, government authorities, and individuals.

Despite its advantages, the system has certain limitations such as dependency on data quality and limited parameters. However, these limitations can be overcome through future enhancements and technological advancements.

Overall, the system provides an efficient, accurate, and scalable solution for air quality monitoring and prediction. It contributes to environmental protection and public health awareness. With further improvements and integration of advanced technologies, the system can play a significant role in building smarter and healthier environments.

FUTURE ENHANCEMENT

The proposed Air Quality Prediction and Pollution Monitoring System provides a strong foundation for real-time monitoring and prediction. However, there are several opportunities to enhance the system further and make it more powerful and widely applicable.

One of the major improvements can be the integration of advanced deep learning models such as neural networks, LSTM (Long Short-Term Memory), and CNN (Convolutional Neural Networks). These models are capable of capturing complex patterns and temporal dependencies in data, which can significantly improve prediction accuracy, especially for time-series air quality data.

Another important enhancement is the integration of IoT-based sensors. By connecting real-time sensors placed in different locations, the system can collect live air quality data continuously. This will make the system more dynamic and useful for real-world applications such as smart cities and environmental monitoring systems.

The system can also be extended by developing a mobile application, which allows users to access air quality information anytime and anywhere. Mobile notifications can be used to send instant alerts when pollution levels become dangerous, increasing user awareness and safety.

Deployment on cloud platforms is another key improvement. Cloud integration will allow large-scale data storage, faster processing, and remote access. It also enables multiple users to access the system simultaneously without performance issues.

The system can be enhanced by including additional environmental parameters such as wind speed, rainfall, atmospheric pressure, and geographical factors. These parameters can provide more accurate and detailed predictions.

Another future enhancement is the use of real-time data visualization dashboards, which provide interactive and dynamic graphs for better analysis. This will help users and authorities understand pollution trends more effectively.

The system can also be expanded for smart city applications, where it can be integrated with traffic control systems, industrial monitoring, and public health systems to reduce pollution levels.

Furthermore, continuous model improvement can be achieved by using real-time data updating and retraining techniques, which allow the system to learn from new data and improve its performance over time.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

References

[1] World Health Organization (WHO), Air Quality Guidelines: Global Update 2021, WHO Press, Geneva, 2021.

[2] U.S. Environmental Protection Agency (EPA), Air Quality Index (AQI) Basics, Available: https://www.epa.gov/aqi

[3] Central Pollution Control Board (CPCB), National Air Quality Index, Government of India, Available: https://cpcb.nic.in

[4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 12, 2011.

[6] McKinney, W., "Data Structures for Statistical Computing in Python," Proceedings of the Python in Science Conference, 2010.

[7] Hunter, J.D., "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, 2007.

[8] Rehurek, R. and Sojka, P., "Software Framework for Topic Modelling with Large Corpora," 2010.

[9] Python Software Foundation, Python Language Reference Manual, Available: https://docs.python.org

[10] NumPy Developers, NumPy Documentation, Available: https://numpy.org

[11] Pandas Development Team, Pandas Documentation, Available: https://pandas.pydata.org

[12] Scikit-learn Developers, Scikit-learn User Guide, Available: https://scikit-learn.org

[13] XGBoost Developers, XGBoost Documentation, Available: https://xgboost.readthedocs.io

[14] Streamlit Inc., Streamlit Documentation, Available: https://streamlit.io

[15] Kaggle, Air Quality and Pollution Datasets, Available: https://www.kaggle.com

[16] Brownlee, J., Machine Learning Mastery with Python, Machine Learning Mastery, 2016.

[17] Bishop, C.M., Pattern Recognition and Machine Learning, Springer, 2006.

[18] Han, J., Kamber, M., and Pei, J., Data Mining: Concepts and Techniques, Elsevier, 2011.

[19] Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.

[20] Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C., and Baklanov, A., "Real-time Air Quality Forecasting, Part I: History, Techniques, and Current Status," Atmospheric Environment, 2012.