International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

Machine Learning-Based System for Weather Prediction and Air Quality Index Estimation

Adnan Mohammed[1], S. Roshan Zameer[2], Umar Chowdhry[3] and Dr. Ashok Kumar[4]
[1]Undergraduate Student, Department of Information Science and Engineering, B.M.S College of Engineering, INDIA
[2]Undergraduate Student, Department of Information Science and Engineering, B.M.S College of Engineering, INDIA
[3]Undergraduate Student, Department of Information Science and Engineering, B.M.S College of Engineering, INDIA
[4]Professor, Department of Information Science and Engineering, B.M.S College of Engineering, INDIA

[1]Corresponding Author: adnan.is20@bmsce.ac.in

Received: 20-03-2024 Revised: 08-04-2024 Accepted: 28-04-2024

ABSTRACT

This paper presents a study on "Machine Learning for Weather Prediction and Air Quality Index Estimation," aimed at enhancing weather forecasting and air quality monitoring. Integrating historical weather data with real-time atmospheric measurements from the OpenWeather API, the study utilizes the Random Forest Machine Learning algorithm to construct predictive models. Backend operations are managed by a Django application on AWS EC2, supported by Nginx as a reverse proxy. The frontend, a ReactJS-based web app hosted on AWS S3 and distributed via CloudFront, offers an intuitive interface. Additionally, a dedicated mobile app extends the system's reach, delivering real-time updates on weather conditions and air quality. This comprehensive approach empowers users with precise insights for informed decision-making and environmental awareness.

Keywords— Air Quality Index, Django, Random Forest, Weather Prediction, CloudFront, ReactJs, Webview

I. INTRODUCTION

Weather forecasting is vital across sectors, safeguarding lives, property, and economic activities from the adverse impacts of extreme weather events such as storms, floods, and heatwaves [1]. However, accurately predicting weather conditions remains a complex and challenging task, especially considering the rapid shifts in environmental conditions due to climate change [4, 18]. The field of meteorology heavily relies on precise weather forecasts to ensure safety planning, effective industry operations, and informed economic decision-making [6, 9]. For example, agricultural forecasts, which rely on factors like temperature, humidity, wind, and precipitation outlook, are crucial for crop planning, trading in agricultural markets, and ensuring food security [1]. Similarly, utility companies use temperature forecasts to estimate future energy demands, while individuals and businesses rely on weather forecasts to plan activities and mitigate risks associated with adverse weather conditions [1].

Analyzing vast amounts of meteorological data presents significant challenges, which are addressed through various data mining procedures, including machine learning techniques [1]. Machine learning plays a pivotal role in weather prediction and Air Quality Index (AQI) estimation, establishing critical links between weather patterns and air quality [1, 17]. The correlation between climate, weather, and air quality profoundly impacts human health, with air pollution contributing to millions of premature deaths worldwide [1]. Therefore, weather forecasting and AQI estimation become essential tools in managing and mitigating the adverse health effects of air pollution [1].

Despite the challenges of atmospheric complexity and data accuracy, weather forecasts remain indispensable for safety planning, industry operations, and economic decisions [1]. However, traditional systems for weather prediction and AQI estimation rely on conventional meteorological models, historical data archives, and real-time monitoring techniques [1]. These systems integrate various data sources such as historical meteorological records, real-time satellite imagery, ground-level measurements, and APIs like OpenWeather, Air Pollution, and Weather Map [1].

Traditional systems have advantages, benefiting from comprehensive historical data archives and real-time updates derived from satellite and ground-based monitoring, enhancing prediction accuracy [1]. Additionally, the integration of various APIs enriches the system with diverse and up-to-date insights, improving overall efficacy [1]. Some traditional systems also offer user-friendly interfaces, facilitating easy manipulation and operations for users with varying levels of expertise [1].

However, traditional systems face drawbacks such as time-consuming data analysis and reduced accuracy [1]. They may struggle with adapting rapidly to fast-changing weather patterns and emergent

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

environmental shifts, impacting responsiveness and the ability to predict extreme weather events effectively [1]. Furthermore, the setup process for traditional systems can be complex and resource-intensive, posing barriers to easy deployment and accessibility, thus limiting usability [1]. In emergency situations, where timely predictions are crucial for decision-making and risk mitigation, traditional systems may struggle with real-time responsiveness [1].

II. LITERATURE SURVEY

Recent studies have significantly advanced the integration of machine learning techniques for predicting air quality and weather conditions. Gupta et al. (2023) conducted a meticulous comparative analysis, meticulously evaluating machine learning algorithms such as Random Forest, Support Vector Machines, and Neural Networks for predicting the Air Quality Index (AQI). Their study not only provided insights into algorithm performance but also delved into the potential applications and limitations of each model. By leveraging existing research and gathering data from various air quality parameters, the authors aimed to contribute to the body of knowledge guiding policymakers and environmentalists in making well-informed decisions regarding machine learning models for real-time AQI prediction.

Maltare and Vahora (2023) explored predicting AQI in Ahmedabad City using machine learning techniques. Their study involved meticulous data collection, rigorous feature engineering, diverse machine learning model application, and thorough evaluation. The aim was to identify accurate models and understand factors influencing urban air quality. The findings provide valuable insights for policymakers and environmental organizations addressing urban air pollution challenges.

Ravindiran et al. (2023) delved into predicting air quality levels in Vishakhapatnam, an Indian coastal city, utilizing machine learning models. Their research not only emphasized the superior accuracy achieved by machine learning models in predicting air quality levels but also explored the unique challenges faced by coastal cities in managing and improving air quality. The study provided practical implications for local authorities and policymakers, shedding light on the factors contributing to air quality variations in coastal regions.

Singh et al. (2022) adopted a comprehensive approach by investigating the use of machine learning techniques for air quality prediction, incorporating diverse data sources such as air quality parameters, meteorological data, and environmental factors. Through the application of various machine learning algorithms and rigorous evaluation using relevant metrics, the study aimed to contribute to effective environmental management. Addressing the complexity of forecasting air pollution levels, the research proposed potential solutions to mitigate their impact, thereby enhancing public health and environmental sustainability efforts.

Liang et al. (2020) provided detailed insights into machine learning techniques for predicting air quality, focusing not only on the selection of suitable machine learning models but also on the process of model development and critical steps of training and validation. By catering to the needs of environmental researchers, policymakers, and stakeholders, the study offered practical insights into the effective utilization of machine learning models for accurate air quality prediction.

Arcomano et al. (2020) presented an innovative approach by developing a machine learning-based global atmospheric forecast model, trained and validated using extensive atmospheric data. This model holds promise in enhancing global atmospheric forecasting capabilities, with potential applications in weather prediction, climate modeling, and other fields reliant on accurate atmospheric forecasts. The comprehensive nature of this research contributes to the broader understanding of machine learning applications in atmospheric science, paving the way for advancements in predictive modeling for environmental monitoring and management.

Hennayake et al. (2021) addressed the need for localized and short-term weather predictions by presenting a machine learning-based model. Utilizing meteorological parameters and historical data, their research aimed to demonstrate the potential of machine learning in providing dynamic insights for short-term weather forecasts, particularly in regions with unique climatic conditions such as Sri Lanka. This study fills a crucial gap by focusing on the specific requirements of localized weather prediction, offering practical applications for regions with distinct weather patterns and environmental challenges.

Li et al. (2021) continued the exploration of machine learning techniques for air quality prediction, emphasizing the importance of data collection, model development, and evaluation. By addressing challenges and proposing a more efficient method for forecasting air pollution levels, the study contributes to ongoing efforts in environmental monitoring and management, ultimately aiming to enhance decision-making processes by providing accurate and timely air quality assessments.

Méndez et al. (2023) took a holistic approach by conducting a survey of machine learning algorithms used in air quality forecasting. By categorizing and evaluating various models, their research provided a comprehensive overview, acting as a guide for researchers, environmental scientists, and policymakers in selecting appropriate algorithms for specific scenarios. This survey contributes to the collective understanding of machine learning applications in air quality prediction, offering insights into the strengths and limitations of different algorithms and

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

informing decision-making in environmental monitoring and management.

Singh et al. (2019) conducted a study investigating the potential of machine learning algorithms to enhance weather forecasting accuracy. Their research delved into various aspects, including algorithm selection, model evaluation, and advancements in weather prediction techniques. By analyzing these factors, the study shed light on the evolving role of machine learning in meteorological sciences and its contributions to improving forecast reliability.

Cican et al. (2023) investigated machine learning methods for predicting air quality in Bucharest City, emphasizing data collection, model implementation, and validation. The study's findings offer insights for local authorities, policymakers, and environmental agencies in managing air quality issues in urban settings. By addressing the specific challenges of air quality prediction in an urban environment, this research contributes to the development of tailored solutions for densely populated areas, thereby supporting efforts to mitigate air pollution and improve public health outcomes.

Reddy et al. (2021) demonstrated the effectiveness of supervised learning techniques in predicting air quality parameters. By addressing key areas such as algorithm selection, model evaluation, and practical applications, their research contributed to understanding supervised learning methodologies in air quality forecasting. The study provides insights into the applicability of different supervised learning models for accurate and reliable air quality predictions, supporting efforts to enhance environmental monitoring and management.

Reilly (2023) discussed the application of machine learning techniques to improve weather forecast accuracy. By highlighting the role of machine learning algorithms in weather prediction models, the article underscored potential advancements in forecasting and their practical applications across various sectors reliant on accurate weather forecasts. The research emphasizes the continuous evolution of weather prediction methodologies and the role of machine learning in shaping the future of meteorological sciences.

Janarthanan et al. (2021) explored deep learning's application in predicting urban Air Quality Index (AQI). Their study detailed data collection, model development, and validation, offering insights into deep learning's effectiveness in urban air quality prediction.

Chantry et al. (2023) provided a comprehensive view of machine learning's role in weather forecasting, discussing its evolution and potential. Their research highlights machine learning's ability to enhance forecast accuracy, benefiting sectors reliant on precise weather predictions.

In summary, while previous studies have made significant contributions to advancing machine learning in environmental monitoring, there remains a need for a more integrated and adaptable system. The proposed system in this study aims to address these gaps by offering a comprehensive solution that enhances weather forecasting precision, AQI estimation, and mitigates existing limitations in the literature. Through a holistic approach, our system seeks to advance environmental monitoring, decision-making, and public health management, supporting efforts to mitigate environmental risks and improve overall environmental quality. The objectives of our study are as follows:

1. Develop robust predictive models utilizing historical weather data and real-time atmospheric measurements through the Random Forest Machine Learning algorithm.
2. Implement a comprehensive system comprising a Django backend and a responsive ReactJS-based web application hosted on AWS infrastructure, with extended functionality to mobile devices.
3. Enhance AQI estimation using tailored machine learning algorithms.

III. DESIGN METHODOLOGY

A. System Requirements

The proposed system aims to enhance weather forecasting and AQI estimation through advanced machine learning and comprehensive data processing. Functional requirements include developing cutting-edge machine learning models for precise weather prediction and robust AQI estimation. Real-time access to diverse weather data sources through APIs ensures timely and relevant updates, with inclusive design for historical data retrieval. The system emphasizes air quality assessment with rigorous data collection and validation mechanisms. Non-functional requirements prioritize robustness, security, scalability, and compliance with industry standards. Hardware requirements include resilient infrastructure capable of handling data-intensive tasks. Visual Studio Code, Django, and Python 3.7 form the software stack.

B. System Architecture

The proposed system architecture efficiently manages the intricacies of weather forecasting and air quality estimation through a modular design. Each module serves a specific purpose, encompassing critical functionalities such as data collection, prediction algorithms, user interface components, and API integration. This modular organization promotes maintainability and adaptability by minimizing dependencies between modules, allowing for smoother modifications and updates. Furthermore, the modular design approach enhances component reusability, facilitating seamless integration of machine learning algorithms into various applications or future enhancements. This design

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

strategy aligns with scalability, enabling the system to efficiently adapt to changing needs or expanded functionalities.

At its core, the system aims to create a robust platform to meet diverse user needs for comprehensive weather information. It encompasses detailed forecasts, real-time temperature variations, and various meteorological data points to offer users an all-encompassing weather experience. Notably, the inclusion of speech recognition capabilities enhances accessibility, especially for older demographics.

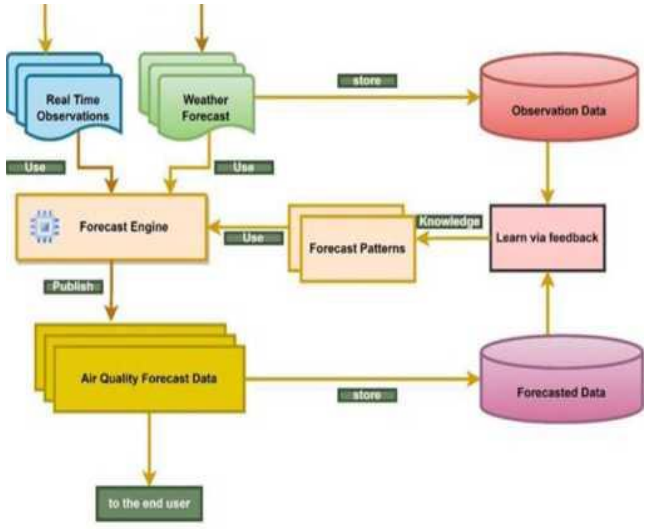

Figure 1: System Architecture

The system relies on an Application Programming Interface (API) to efficiently collect and process real-time weather data from multiple reliable sources. This integration ensures a broad spectrum of meteorological insights, covering current conditions and future forecasts. Users can receive tailored and precise weather updates for their current location or specific cities of interest. Behind the scenes, sophisticated algorithms guarantee the accuracy and reliability of the provided weather information. By processing extensive datasets and leveraging cutting-edge predictive models, the system generates highly accurate forecasts and real-time updates. This meticulous approach enhances the system's reliability and credibility, ensuring users receive precise and up-to-date weather information.

C. Algorithms

1). Algorithm for Weather Report:

STEP 1: Import libraries specific city once enter the city name.
STEP 2: Using API can predict the weather report for current specific city once enter the city name.
STEP 3: API URL: "https://api.openweathermap.org/data/2.5/weather?q="+city+"&appid=06c921750b9a82d8f5d1294e1586276f`
STEP 4: Using the Json to get data STEP 5: Display the output.

2) Random Forest Algorithm for Temperature Prediction:

STEP 1*: Loa*d the dataset into the python and import the libraries
STEP 2: Data pre-processing
Step 2.1: Drop the unnecessary column from the dataset.
Step 2.2: Delete all values from te pressure which has a value -9999.
Step 2.3: Taking all the features into x variable and y for prediction.
Step 2.4: Set the dummies value as a level for the weather classification.
Step 2.5: Delete last dummies value which is null.
Step 2.6: Concatenate the dummies value with the input feature X.
Step 2.7: Created the new dataset after apply the preprocess.
STEP 3: Train and test the data
Step 3.1: Splitting Dataset into train set and test set.
STEP 4: Fitting the Random Forest Regression model.
STEP 5: Histogram of data (visualization)
STEP 6: Analyze the original with predicted data (using visualize).
STEP 7: Output.

3) *Algorithm for Rainfall Prediction*

STEP 1*: Loa*d the data and import libraries
STEP2: Replace the value for 'RainToday' and 'RainTomorrow' columns with (No:0, yes:1)
STEP 3: Visualize the data
STEP 4: Output.

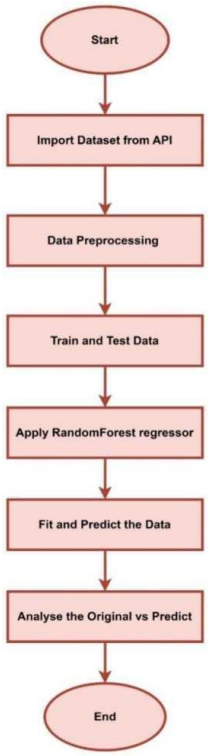

Figure 2: Flow Chart

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

D. Class, Activity and Use Case Diagrams

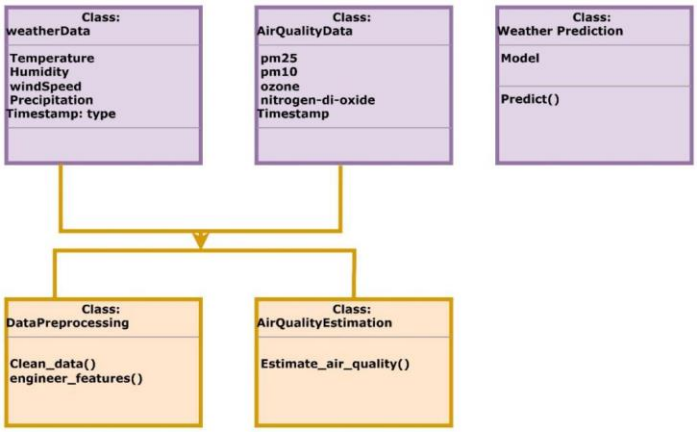

Figure 3: Class Diagram

The class diagram depicted in Fig 3 serves as a foundational tool for visualizing the structure and relationships between classes within the system architecture. In the context of our study objectives and system architecture, the class diagram depicts essential classes such as weatherData, representing key meteorological parameters like temperature, humidity, wind speed, and precipitation. These classes encapsulate attributes and methods necessary for processing and analyzing weather data. Relationships between classes illustrate associations and dependencies crucial for the system's functionality, facilitating design visualization and communication among stakeholders.

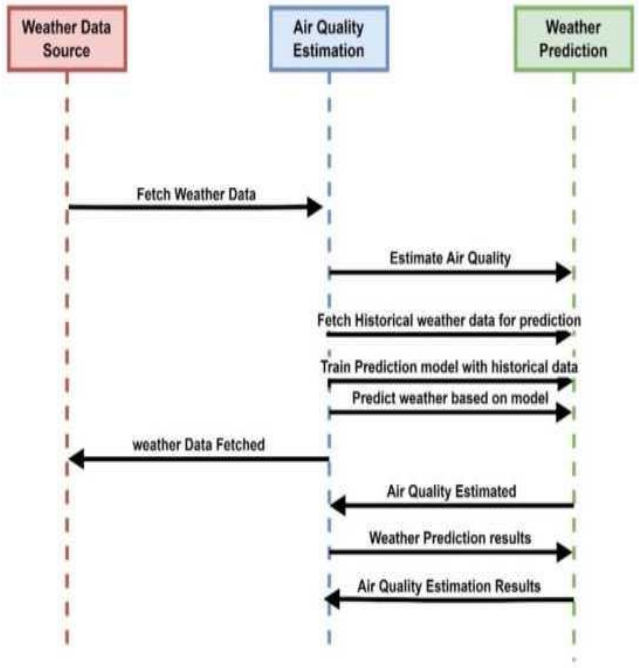

Figure 4: Activity Diagram

The activity diagram shown in Fig. 4 provides a graphical representation of the flow of activities within the system, aligning with our study objectives and system architecture. Activities within the diagram correspond to essential processes and functionalities related to weather forecasting and air quality estimation. Transitions and decision points depict the sequence of operations and decision-making processes involved in data processing and analysis. By visualizing the flow of activities, the activity diagram aids in understanding the system's architecture, communication flow, and coding structure, serving as a blueprint for implementing the system's functionalities.

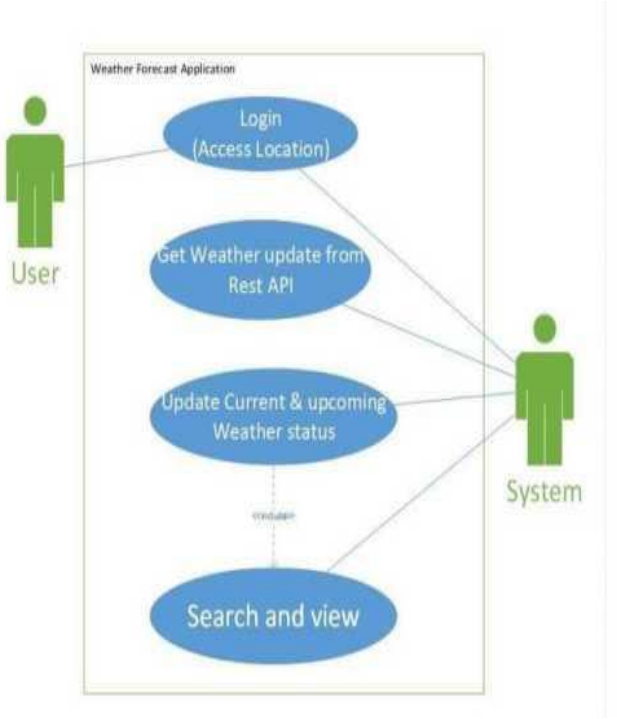

Figure 5: Use-Case Diagram

The Use-case diagram depicted in Fig. 5 offers a top-level portrayal of a system's functionality and scope, aligning with the study objectives. They visually articulate the interactions between system components and external actors, reflecting how users engage with the system to achieve specific objectives. While they illustrate what the system does and how actors utilize it, they do not delve into the internal operations outlined in the system architecture.

IV. SYSTEM IMPLEMENTATION

A. Process Overview

The implementation of the proposed ML system followed a structured process to seamlessly integrate various components for accurate weather forecasting and air quality assessment:

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

1) *Data Acquisition and Preprocessing*: Weather and AQI data were obtained from the OpenWeather API and preprocessed for consistency.
2) *Machine Learning Model Development*: Random Forest models were developed using historical data for weather prediction and AQI estimation.
3) *Backend Infrastructure Setup*: The system was deployed on AWS EC2 with Django and Nginx for real-time forecasting.
4) *Frontend Development*: A user-friendly web interface was created using ReactJS, hosted on AWS S3 and CloudFront.
5) *Mobile Application Development*: A mobile app was developed using a WebView-based approach for cross-platform consistency.
6) *Testing and Validation*: Rigorous testing ensured reliability, accuracy, and performance of the system.
7) *Deployment and Monitoring*: The system was deployed into production, with real-time monitoring for tracking performance.

This structured approach facilitated the successful implementation of a comprehensive solution for weather prediction and air quality monitoring.

B. Backend Infrastructure and its Deployment

The backend infrastructure of the proposed ML system is anchored by a Django application deployed on AWS EC2, tightly integrated with the Random Forest machine learning algorithm for precise weather forecasting and AQI estimation. Nginx operates as a reverse proxy, facilitating efficient request routing, load balancing, and heightened security, forming a resilient foundation for the weather prediction and AQI estimation system. Within the Django application, various components collaborate to manage user requests and process data efficiently. Views, represented by Python functions or classes, handle incoming requests and formulate appropriate responses, orchestrating the flow of data between the user interface and backend logic. Additionally, models in Django define the data structure of the application and interact with the database. These models encompass classes responsible for storing user information, weather data, AQI measurements, and other pertinent data. Moreover, the Django application seamlessly integrates the Random Forest machine learning model, leveraging libraries like scikit-learn, to analyze historical weather data and current atmospheric measurements for forecasting future weather conditions and estimating the AQI.

To optimize performance and security, Nginx serves as a reverse proxy in front of the Django application. It efficiently manages incoming requests, distributes the load across multiple backend servers if necessary, and enhances security by acting as a barrier between the internet and the Django application. Furthermore, Nginx boosts performance by directly serving static files, relieving the Django application from this task and enabling it to focus on processing dynamic requests. This configuration optimizes resource utilization and ensures a seamless user experience, even during peak traffic periods.

The backend deployment process encompasses a series of essential steps to ensure the seamless operation of the ML-based weather prediction and AQI estimation system. Initially, the deployment procedure begins with the launch and configuration of an AWS EC2 instance, where the Django application will reside. This involves selecting an appropriate instance type, configuring security groups to control inbound and outbound traffic, and setting up any necessary networking configurations.

Once the EC2 instance is provisioned, the next step involves deploying the Django application onto the instance. This process typically entails uploading the application code to the EC2 instance, either manually or through version control systems like Git. After the code is uploaded, a virtual environment is set up within the EC2 instance to isolate the application's dependencies from other system packages. This ensures that the application can run independently with its required libraries and packages.

Following the setup of the virtual environment, the next crucial step is installing dependencies necessary for the Django application to function correctly. These dependencies may include Django itself, as well as any additional Python packages required for the application's specific functionality, such as database connectors, web frameworks, or machine learning libraries.

Once all dependencies are installed, the Django application is ready to be served to users. To achieve this, a WSGI (Web Server Gateway Interface) server like Gunicorn is configured to run the Django application within the virtual environment. Gunicorn acts as a bridge between the Django application and the outside world, handling incoming HTTP requests and serving responses generated by the Django application.

Finally, Nginx is configured to act as a reverse proxy server, forwarding incoming requests to the Gunicorn server. This setup enhances security by shielding the Django application from direct exposure to the internet and allows for more efficient request handling and load balancing. Nginx also provides additional features such as caching, SSL termination, and content compression, further optimizing the performance and security of the deployed application.

Once Nginx is configured, the backend deployment procedure is complete, and the ML-based weather prediction and AQI estimation system is ready to operate smoothly. This comprehensive deployment process ensures that the backend infrastructure is robust, scalable, and capable of handling the demands of real-world usage scenarios.

C. Frontend Infrastructure and its Deployment

The frontend infrastructure of the proposed ML system was designed to provide users with a responsive and intuitive interface using ReactJs, a JavaScript library known for its ability to create dynamic web applications. Leveraging React's component-based architecture, the development process emphasized modular design to create reusable UI elements, facilitating efficient and scalable application development. This approach resulted in the creation of a React-based web app, serving as the primary

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

interface for users to interact with weather and air quality information.

To ensure a user-friendly and responsive experience, the ReactJS-based web application was deployed on AWS S3 and distributed via CloudFront. This deployment strategy enabled seamless access to the system across various devices and screen sizes, enhancing accessibility for users.

The frontend interface was carefully crafted to prioritize accessibility and enhance user experience. Key elements such as weather cards visually presented essential weather information, including temperature, humidity, wind speed, and precipitation for selected locations. These weather cards provided users with a quick overview of current weather conditions and forecasts, aiding in activity planning.

Additionally, AQI indicators were integrated into the interface to provide insights into air quality levels, displaying the current AQI value along with corresponding health advisories. This feature empowered users to make informed decisions about outdoor activities and exposure to air pollutants. Moreover, location selectors were seamlessly integrated, allowing users to specify their preferred location for weather and AQI information. Whether through manual input or geolocation services, users could retrieve personalized and relevant data to suit their needs. The interface also offered customization options, enabling users to tailor their experience by selecting between metric and imperial units for weather data, setting preferences for notifications or alerts, and adjusting other display settings to align with their preferences.

The ReactJS web app's deployment took place on AWS (Amazon Web Services), chosen for its reliability and scalability. Utilizing AWS services such as Amazon EC2 for virtual server hosting and Amazon S3 for web hosting, the deployment aimed to establish a robust infrastructure. Deploying the ReactJS application on AWS via S3 and CloudFront offered a cost-effective solution for serving static web content with high availability and performance.

The deployment process involved transitioning the web app from development to production through several essential steps:

1) Ensuring the stability and readiness of the codebase, including necessary configuration changes or updates.
2) Compiling and generating optimized assets and bundles for deployment.
3) Configuring environment variables, database connections, and other runtime settings.
4) Uploading the built application code to the hosting platform.
5) Configuring the server environment.
6) Conducting thorough tests to verify functionality, performance, and compatibility.
7) Setting up monitoring and logging tools for real-time tracking of application performance and issue detection.

This meticulous deployment process facilitated the seamless transition of the ReactJS web app to production, ensuring optimal performance and availability for users accessing weather forecasts and air quality information.

D. Mobile App Deployment

The mobile app development in the present study aimed to create a versatile and user-friendly application delivering real-time weather and air quality information for mobile devices. It served as an extension of the existing web-based platform, ensuring users could access essential forecasts and AQI data on the go. Integrated with backend APIs, the app provided accurate and up-to-date updates seamlessly.

In the mobile app development, a WebView-based approach was adopted for rendering web content within the app. It allowed displaying web content like HTML pages or web applications within a native mobile app environment. This approach reduced development time, ensured consistency across platforms, offered flexibility in updates, and proved cost-effective compared to native app development.

The front-end integration process involved seamlessly incorporating the React-based web application into the mobile app using a WebView component. This component acted as a container within the native mobile app, displaying web content either from a specified URL or a local file path. By carefully configuring this setup, the mobile app was able to integrate the functionality and content of the web application, ensuring a consistent user experience across both platforms.

During development, a JavaScript Bridge facilitated seamless communication between the web and mobile apps, enabling bidirectional interaction for enhanced user experience. Efforts were made to optimize the web app's layout and visual elements for mobile devices, ensuring readability and responsiveness. Special focus was placed on optimizing the mobile app for Android, following Material Design guidelines for visual consistency and implementing performance enhancements for smoother operation across various devices.

V. RESULTS

A. Web App Result

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

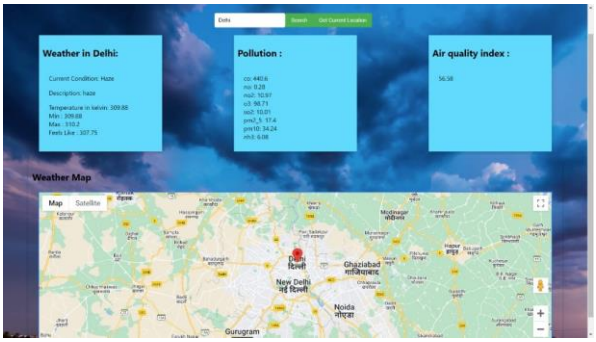

Figure 6: Web App Display

Users access the web app via the URL https://d118448h5e241h.cloudfront.net/, where they find a user-friendly interface featuring a map component. They can select a city either by clicking on its location on the map or by typing its name into a search box. Once a city is selected, real-time weather data and air pollutant concentrations are retrieved using dedicated APIs. The web application displays detailed information on temperature, humidity, wind speed, atmospheric pressure, precipitation, and key air pollutants such as PM2.5, PM10, NO2, SO2, CO, O3, and others. Additionally, a Random Forest Regression algorithm estimates the AQI for the selected city, considering various environmental factors. The output is presented in a clear format, providing users with real-time updates on weather conditions, air pollutant levels, and estimated AQI, enabling informed decisions on outdoor activities and health precautions.

B. Mobile App Result

The mobile app provides users with a detailed overview of current environmental conditions and air quality status for a selected city. Through an intuitive interface featuring a map component, users can choose a city by tapping its location on the map or entering its name into a search box. Once a city is selected, the app fetches real-time weather data and air pollutant concentrations using dedicated APIs. Weather data includes parameters like temperature, humidity, wind speed, atmospheric pressure, and precipitation, while air pollutant concentrations cover key pollutants such as PM2.5, PM10, NO2, SO2, CO, O3, and other relevant gases.

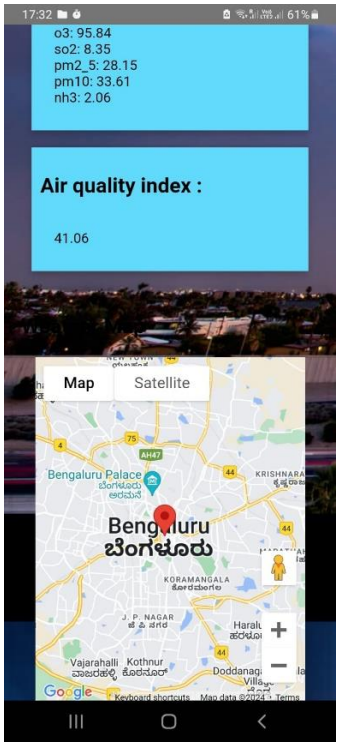

Figure 7: Mobile App Display

VI. CONCLUSION

The paper successfully addressed the need for accurate weather forecasting and air quality monitoring by leveraging machine learning and advanced data analysis. It integrated historical weather data with real-time atmospheric measurements from the OpenWeather API, providing timely forecasts and Air Quality Index (AQI) estimations. The deployment of a Django application on AWS EC2 for processing data, coupled with a ReactJS-based web app on AWS S3 and CloudFront for user access, demonstrates a robust and scalable infrastructure. Furthermore, extending this functionality to mobile devices through a dedicated mobile app using a WebView-based approach ensures accessibility and usability across different platforms.. Future enhancements include improving disaster prevention, speech recognition, and integrating advanced algorithms like deep learning. Strengthening information dissemination and fostering collaborative research will further enhance the present study's impact on environmental monitoring and resilience.

Peer Reviewed & Refereed Journal
Volume-14, Issue-2 (April 2024)
https://doi.org/10.5281/zenodo.11084950

REFERENCES

[1] S. Gupta, Y. Mohta, K. Heda, R. Armaan, B. Valarmathi & A. Ganeshan. (2023). Prediction of air quality index using machine learning techniques: A comparative analysis. *Journal of Environmental and Public Health, 2023*, 1-26. DOI: 10.1155/2023/4916267.

[2] N. Maltare & S. Vahora. (2023). Air quality index prediction using machine learning for Ahmedabad city. *Digital Chemical Engineering, 7*, 100093. DOI: 10.1016/j.dche.2023.100093.

[3] G. Ravindiran, T. G. Hayder, K. Kanagarathinam, A. Alagumalai & C. Sonne. (2023). Air quality prediction by machine learning models: A predictive study on the indian coastal city of Vishakhapatnam. *Chemosphere*. DOI: 10.1016/j.chemosphere.2023.139518A.

[4] R. Singh, S. Raghav, T. Maini, M. Singh & Md. Arquam. (2022). Air quality prediction using machine learning. *SSRN Electronic Journal*. DOI: 10.2139/ssrn.4157651.

[5] Y.-C. Liang, Y. Maimury, A. H.-L. Chen & J. R. Cuevas Juarez. (2020). Machine learning based prediction of air quality. *Applied Sciences, 10*(24), 9151.

[6] T. Arcomano, I. Szunyogh, J. Pathak, A. Wikner, B. Hunt & E. Ott. (2020). A machine learning - Based global atmospheric forecast model. *Geophysical Research Letters, 47*. DOI: 10.1029/2020GL087776.

[7] K. Hennayake, D. Dinalankara & D. Mudunkotuwa. (2021). Machine learning based weather prediction model for short term weather prediction in Sri Lanka. *Proceedings of the IEEE Conference IEEE ICIAfS 2021 - Endowing Intelligent Sustainability*. DOI: 10.1109/ICIAfS52090.2021.9606077.

[8] Li, Y. Li & Y. Bao. (2021). Research on air quality prediction based on machine learning. *2nd International Conference on Intelligent Computing and Human Computer Interaction (ICHCI), Shenyang, China*, pp. 77-81. DOI: 10.1109/ICHCI54629.2021.00022.

[9] M. Mendez, M. Merayo & M. Nunez. (2023). Machine learning algorithms to forecast air quality: A survey. *Artificial Intelligence Review, 56*, 1-36. DOI: 10.1007/s10462- 023-10424-4.

[10] S. Singh, M. Kaushik, A. Gupta & A. K. Malviya. (2019). Weather forecasting using machine learning techniques. *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*.

[11] G. Cican, A.-N. Buturache & R. Mirea. (2023). Applying machine learning techniques in air quality prediction—A Bucharest city case study. *Sustainability, 15*(11), 8445.

[12] K. C. Reddy, K. N. Reddy, K. B. Prasad & P. S. Rajendran. (2021). *The prediction of quality of the air using supervised learning. 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India*, pp.1-5. DOI: 10.1109/ICCES51350.2021.9488983.

[13] J. Reilly. (2023). Using machine learning for accurate weather forecasts. Available at: https://www.akkio.com/post/weather-prediction-using-machinelearning. (Retrieved on 2/1/2024).

[14] R. Janarthanan, P. Partheeban, S. Somasundaram & P. Elamparithi. (2021). A deep learning approach for prediction of air quality index in a metropolitan city. *Sustainable Cities and Society, 67*, 102720. DOI: 10.1016/j.scs.2021.102720.

[15] M. Chantry, Z. B. Bouallegue, L. Magnusson, M. Maier-Gerber & J. Dramsch. (2023), *The rise of machine learning in weather forecasting*. Available at: https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machinelearning-weather-forecasting. (Retrieved on 5/1/2024).