International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

AIR MAP- Deep Learning Prediction in Air Quality for Smarter Decisions

Madduri Tejaswi[1], Ravipati Bavika[2], Sarvadey Sakshi[3], Yarva David[4], Ch Shanti Priya[5]

[1]*Computer Science and engineering, Hyderabad Institute of Technology and Management*
[2] *Computer Science and engineering, Hyderabad Institute of Technology and Management*
[3] *Computer Science and engineering, Hyderabad Institute of Technology and Management*
[4] *Computer Science and engineering, Hyderabad Institute of Technology and Management*
[5] *Computer Science and engineering, Hyderabad Institute of Technology and Management*

Abstract - Air Map is a forward-thinking web application which provides customers with up-to date information about air quality and weather forecasting. Pollutant levels and weather are predicted via deep learning model using historic data extracted carefully from sensors. This dataset undergoes data pre-processed to ensure its accuracy and trained by deep learning model on this vast data to predict the future parameters with high accuracy. These trends can be better perceived by using visualization tools such as time-series plot and heat maps for better understanding for customers. This application can be customized to receive timely notification on percentage of hazardous gases and total Air Quality index up to time. This application act as a catalyst for environmental awareness for both public and government, by this forecasting citizens will understand the air quality and weather patterns and for government this visualization will help to understand the weather and air quality patterns to mitigate air pollution by applying norms. This application works towards a healthier and more sustainable future.

Key Words: Air Quality Index, Sensor Data, Deep Learning Models, LSTM, Statistical analysis and metrices, Air Map integration

1.INTRODUCTION

Air pollution poses a significant role in threat of the mankind. We breathe in a vast amount of air everyday which triggers various respiratory problems and heart diseases and even cancer. As world is moving ahead rapidly with many automated machines and in busy cities in an average more than 50,000 vehicles move in an hour which releases harmful gases like Carbon monoxide, by predicting and understanding air quality will help netzines to choose indoor on high pollution days, this helps in protecting their health by stopping them to stay outdoor unnecessarily. This air pollution causing gases includes PM2.5, PM10, Nitrogen oxide, Sulphur dioxide, Carbon monoxide, Ozone etc., predicting these gases helps in saving environment and understand Air Quality Index and helps in taking measures and mitigating pollutants level in atmosphere. These checks act as a vital line of defense against multitude of invisible threats suspended in the air we breathe. Our project spearheads a novel approach to air quality prediction by leveraging Deep Learning and visualization tools. A sophisticated Deep Learning model analyzes historic data of air quality and meteorological data collected from the nodes of sensor network for forecasting pollution and meteorological data. This key innovation integrates crucial environmental parameters like PM2.5, PM10, NO2, SO2, O3, CO, Temperature, Pressure and Humidity. This synergy provides a comprehensive understanding of the complex interplay between weather and air quality. Users will be empowered with a user-friendly interface.

2. LITERATURE REVIEW

Air pollution exposure has become one of a significant reason nowadays for heart diseases and premature deaths. To overcome this the increased accuracy using deep learning models equips officials to ensure the level and range of pollutants present around us. Researchers have proposed various ml models such as random forest, neural networks, support vector machines but among other models LSTM model leverages sequential air quality prediction for superior high air quality prediction and temporal dependencies. Evaluating the model performance is undergone by exploring various geographical regions records and evaluation metrics. Evaluating model includes RMSE, MAE methods and correlation indices. By separating the training and testing data RMSE emphasizes large errors by squaring them before taking average and takes outputs, lower values depict the levels are not harmful and can be laid. The selection of hyperparameters are also essential for model formation which entices the training process, overfitting and prediction accuracy. GA (genetics Algorithm) is used

for two crucial tuning parameters such as window size, number of LSTM units, learning rate and optimizer settings. Window size depicts the length of historic data and the units depicts number of parameters taken such as (PM2.5, PM 10, CO, NO2, SO2, ozone). To maintain diversity and search spaces mutation is done for the off spring solutions to avoid sub optimal solutions and repetitive null values. LSTM cells are the core components for recurrent neural networks where they offer a powerful tool for sequential data modelling. The memory cell typically consists of 4 components such as input gate, output gate, forget gate, memory gate which allows to capture data and store past historic data sets. This helps our application to keep accurate forecasts by removing temporal patterns.

2.1 EXSITNG MODELS

All existing models' approaches are hybrid deep learning approaches that incorporate factors influencing air quality and these models' collect data from the various websites and Kaggle. All models utilize combination of 1D Convolutional Neural Network for extracting local trends and some models only calculate AQI index without predicting concentration numeric value of pollutants. These models do not incorporate integration of deep learning model with web frameworks for better visualizations.

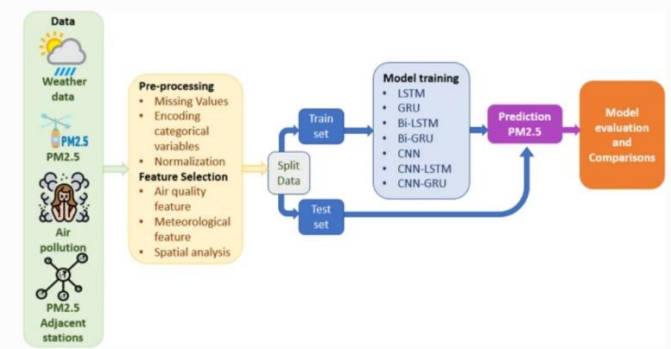
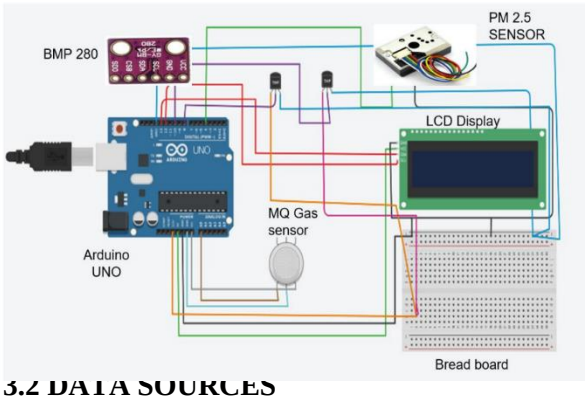


Fig 1: Existing Model flow graph

3. METHODOLOGY

3.1 SENSORS DETAILS

This project aims to mitigate pollution level by employing 3 main sensors importantly such as PM2.5, BMP 280 and MQ series gas sensors. Data is collected from these sensors for enhancement of processing air quality and predicting clear air. PM2.5 sensor is used for measuring air borne particles such as dust and pollens present in air and has diameter up to 2.5 to 10 mm. BMP 280 sensor acts as a barometer capturing atmospheric pressure data and prediction of movement and MQ series sensor with various genres like MQ 2,3,7,135 where gases such as methane, ethanol, carbon monoxide, ammonia, alcohol are detected.



3.2 DATA SOURCES

Various Data streams are harnessed to access and permit mitigated data. Collaboration with Agencies such as TSPCB (Telangana state pollution control board) can be used to unlock various network data and meteorological parameters. The data set consists of temperature, humidity and wind patterns which are then rigorously interpolated, processed for outlier detection.

No	Year	Month	Day	Hour	PM2.5	PM10	O3	SO2	NO2	CO	TEMP	HUM	PREP
1	2023	11	1	0	70	145	4	0	9	840	19.31	92.25	0
2	2023	11	1	1	70	145	4	0	9	849	18.88	94	0
3	2023	11	1	2	71	145	4	0	9	840	18.51	95.06	0
4	2023	11	1	3	71	145	4	0	9	841	18.19	95.38	0
5	2023	11	1	4	71	145	4	0	10	841	18.06	94.56	0
6	2023	11	1	5	71	145	4	0	10	841	17.9	93.75	0
7	2023	11	1	6	71	145	4	0	10	841	18.61	90.62	0
8	2023	11	1	7	72	146	5	0	10	845	20.89	78.44	0
9	2023	11	1	8	72	146	5	0	10	845	23.93	64.81	0
10	2023	11	1	9	72	146	5	0	10	845	25.94	58	0

Fig 3: Sample gathered data

3.3 DATA FORMATTING

After collecting data, they are stored in XLSX format which allows to categorize the parameters accordingly. This format allows for built in data visualization and has larger file size than other data formats. To train neural network hourly data was used from the sensors which is collected at one place and the collected XLSX format is converted to CSV format which is easy to import in deep learning models. Then missing values are replaced with zeroes(fill) and data was scaled with 1 and 0 with min-max scaler.

3.4 OUTLIERS DETECTION AND TREATMENT

Outliers can skew the results of analysis and leads to inaccurate data conclusions which can worsen the whole data set and effecting results. To overcome these IQR (interquartile range) method identifies the outliers, calculates median of lower and upper half of domain data and arranges first quartile to subtract from third quartile. Winsorization is another method used in specific cases for replacing outliers to nearest value for central tendency of data.

3.5 DATA STANDARDIZATION

LSTM models process information by comparing with different equations for better prediction. One such methods include z-score standardization and min-max scaling. Z score standardization is done by subtracting the mean (μ) of each feature from its individual values (x) and then divides by the standard deviation (σ) of that feature. This transformation centers the data around a mean of 0 and scales it to have a standard deviation of 1. Min-max scaling is done by segmenting scales each feature to a specific range, typically between 0 and 1 (or -1 and 1). Here, min(X) represents the minimum value in the feature, and max(X) represents the maximum value. Z-score is generally preferred for LSTMs because it preserves the original distribution of the data, which can be crucial for capturing outliers and extreme values that might significantly impact air quality.

Z-Score Standardization

$Z = (x - \mu) / \sigma$

Where,

- Z is Standardized value
- X is the original value of the data point
- μ is the mean of the feature
- σ is the standard deviation of the feature

Min-Max Scaling

$X\_scaled = (x - min(X)) / (max(X) - min(X))$

Where,

- X_scaled = Scaled values
- X = original value of the data point
- Min(X) = minimum value in the feature
- Max(X) = maximum value in the feature

4. LSTM MODEL

4.1 INTRODUCTION TO LSTM

Long Short-Term Memory or commonly called LSTM is part of Recurrent Neural Network (RNN). This type of architecture is specifically designed to handle sequential and time series data. Unlike traditional neural network which struggle to deal with long-term dependencies, LSTM poses a unique architecture where it remembers the information of the past input. This trait of LSTM model makes it competent to deal with time series forecasting to predict Air Quality and Weather data.

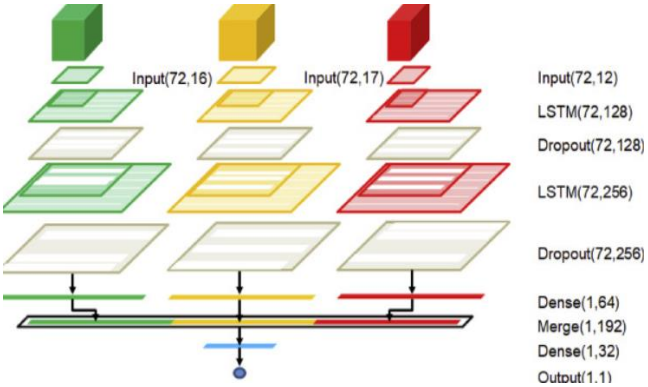


Fig 4: Core structure of LSTM

4.2 LSTM ARCHITECTURE

LSTM Architecture or Memory cell helps us to understand the flow of the model. LSTM model comprises with several key elements that engineer the flow and process given input. The layer of LSTM are Cell State (c), Forget Gate (f_t), Input Gate(i_t), Output Gate (o_t).

- **CELL STATE (C)** This cell acts as the core memory compartment, where essential information from past steps is stored. Unlike standard neuron's activation state, the cell state can persist for extend duration within the network.
- **FORGET GATE (f_t)** This gate serves as a selective filter, deciding what information from the cell state (C_t-1) at the previous time step (t-1) should be retained. It analyzes the previous cell state and the current input (X_t) to generate input vector (i_t). Additionally, it creates a candidate memory cell value (C_t ^ ~) , which

represents the new information that could potentially be added to the cell state.

- **INPUT GATE (i_t)** This gate controls the flow of new information into the cell state. It considers the current input (X_t) and the previous state (C_t-1) to generate input vector. The new information can be added potentially into Input cell.
- **OUTPUT GATE (o_t)** This gate acts as the final checkpoint, determining information from the current state (C_t) will influence the network's output (h_t). It analyzes the current state (C_t) and the previous cell state to generate an output vector (o_t).

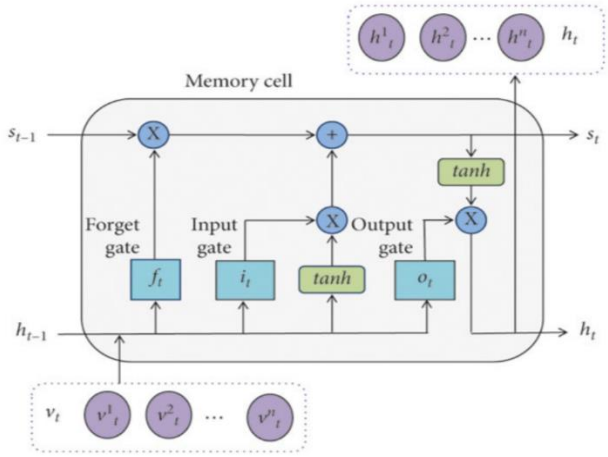


Fig 5: LSTM Memory cell

4.3 STACKED LSTM ALGORITHM

Stacked Long Short-Term Memory Algorithm networks are powerful Deep Learning approach for predicting Air Quality and Weather. Stacked LSTM captures and analyzes complex relationships and long-term dependencies within the data. Stacking up layers in LSTM allows to enhance the model's learning capability.

Stacked LSTM Architecture

A Stacked LSTM Architecture for Air Quality and weather prediction involves

- **Input Layer** This layer receives the preprocessed Air Quality and Weather data
- **Stacked LSTM Layer** The data fed through multiple LSTM layers, with each layer progressively extracting more complex features and dependencies from the data.
- **Output Layer** The final layer takes the output from the last LSTM layer and generates the predicted air quality values or weather conditions for the desired future data.

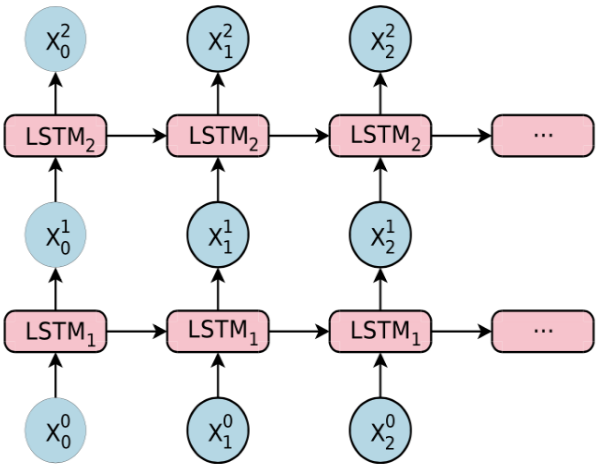


Fig 6: Stack LSTM structure

4.4 MODEL TRAINING

The training stage is heart of LSTM model for air quality and weather prediction. This involves splitting the dataset into three different sets which includes Training set, Validation set, and Testing set.

THE TESTING SET

This is considered as the main character of any Deep Learning model. This helps model to learn things from the given data set. It comprises a significant portion of around 70% to 80% of preprocessed data. The model will be exposed to this data repeatedly, which will adjust its internal parameter to improve prediction accuracy.

THE VALIDATION SET

This set of dataset acts as an observer, typically around 10% to 20% of dataset. The model's performance on this set is monitored throughout the process. This set acts as an early warning to prevent over and under fitting, over fitting and Under fitting are the situation where model memorizes the training data too well or the model does not memorize the training data but these conditions always fail to generalize to unseen data.

THE TESTING SET

This acts as a judge which represents the remaining 10% to 20% of dataset. The model's performance on this unseen dataset provides a final evaluation of its generalizability where it predicts air

quality and weather for the situations that hasn't recognized during training.

FLOW CHART

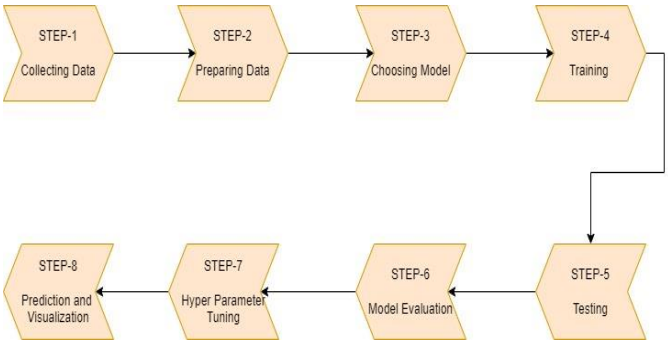


Fig 7: Work flow of our model

4.4 HYPER PARAMETER TUNING

The initial choice of hyper parameters includes number of layers, number of units, learning rate, batch sizes which has significant impact on model's performance

LEARNING RATE

This controls network adjustments like internal weights and biases during training. A high learning rate can lead to rapid improvement but also instability, while a low learning rate might cause slow progress or get struck in sub optimal solutions

NUMBER OF EPOCHS

An epoch represents one complete pass-through entire training. The number of epochs determines how long the network trains. Too few epochs might lead to underfitting and, while too much epochs might lead to overfitting.

BATCH SIZE

This defines the number of data points processed by the network in one training step. A larger batch size can improve training speed but might lead to less precise updates, while smaller batch size can be slower but more accurate.

5. AIR MAP DEVELOPMENT

5.1 FRONT END FRAMEWORK

A front-end framework is a collection of pre-written code and tools that developers use to build a User Interface (UI) and User Experience (UX) of a web application or website.

Given focus on data visualization and user interaction with the Air Map, HTML and Java Script provides a good balance between functionality and development act. For projects with moderate level of complexity like Air Map, this approach can be efficient. HTML and Java script integrates seamlessly with various back-end technologies. The combination of HTML and java script is compatible with many APIs like Flask, Django and Rest

HTML (Hyper Text Markup Language)

HTML is the main foundational language for any web applications or web pages. It defines both content and structure of any web page. This HTML language is built up with tag lines like heading tag, body tag, div class and many more, which acts like a skeleton for any web page. HTML provides basic structure like layout, class definition data display and text elements.

CSS (Cascading Style Sheets)

CSS controls the visual presentation of any web page. It defines styles, fonts, colors, background and positioning of the elements created with HTML and enhances the overall appearance of the web page. Air Map components like appearance, pollution level color coding is defined by CSS

JAVA SCRIPT

Java script is the versatile programming language which adds dynamic behavior to the web pages. It allows you to create animations, response to the user actions like clicks and scrolling and manipulation of the elements written in HTML script. Java script framework offers many pre built components, tools and functionalities to streamline web development. They often enforce specific coding structure and provide features for managing data flow, handling user interaction and building complex UIs, Java script framework also allows to fetch and call APIs for functioning of the web pages.

5.2 BACK-END FRAMEWORK

Back-end framework is a collection of tools and libraries that developers use to build the server-side logic of web application. It acts as the foundation for the behind-the-scenes functionality that user doesn't directly interact with. The main functionality of back-end

frameworks are Data Management, Request Handling, Business Logic Implementation, Security Features, Server-side scripting.

FLASK

Flask is a light weight and versatile web framework written in python. It provides foundation for building web application by handling Routing i.e., It defines the route that map URLs to the specific functions with in the application. Flask handles Request handling i.e., It processes incoming request from the user's browser and Response Generation i.e., Flask generates dynamic content from HTML and JSON on user's request and application logic.

5.3 INTEGRATION

Deep Learning library Tensor Flow and Flask work seamlessly when you train any deep learning model the data log and out log are saved in Tensor flow when you import these libraries while training your model. TensorFlow acts as a internal database for mobile/resource-constrained environments.

Routes are defined in Flask application using decorators. This route will handle user requests for air quality and weather forecasting. When user interacts with the web application, Flask can the user's input through request.args function, this includes prediction data. Within Flask route function we need to load the previously saved Tensor Flow model using libraries like tensorflow.saved_model or tf.keras. Once model generates air quality prediction and weather forecast based on the learned patterns from the historic data Flask allows us to generate predicted air quality and weather parameters in JSON format data for easy integration with front-end integration.

6. VISUALIZATION AND RESULT

Visualization plays crucial role in creating visual representation of information. It transforms complex data into easily understandable charts, graphs, maps and other visual elements. Visualization helps us to understand the patterns and correlation between the parameters of the data trained.

Time Series Plots

Time series Plots historical air quality and weather data over time and day to reveal the trends and seasonality. This helps to visually compare between Predicted vs Actual parameters.

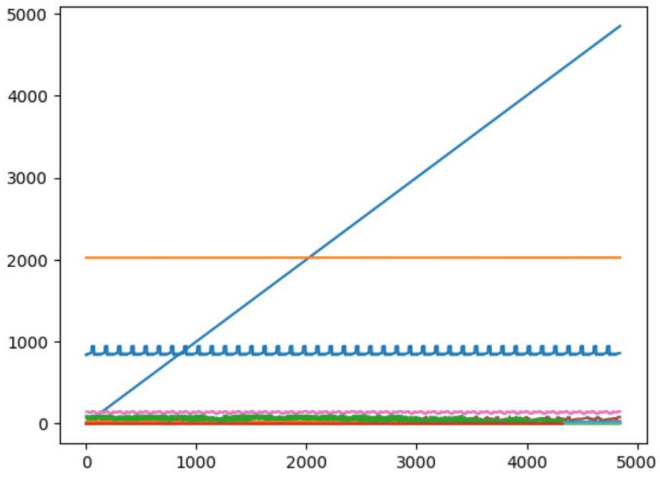


Fig 8: Timer series plot between all Parameters

Scatter Plots

Scatter plot explores relationship between two parameters for instance Air quality parameter vs Air Quality parameter or between Air quality and Weather forecast parameter. Scatter plot color-codes the data points based on predicted vs actual values to identify potential prediction biases.

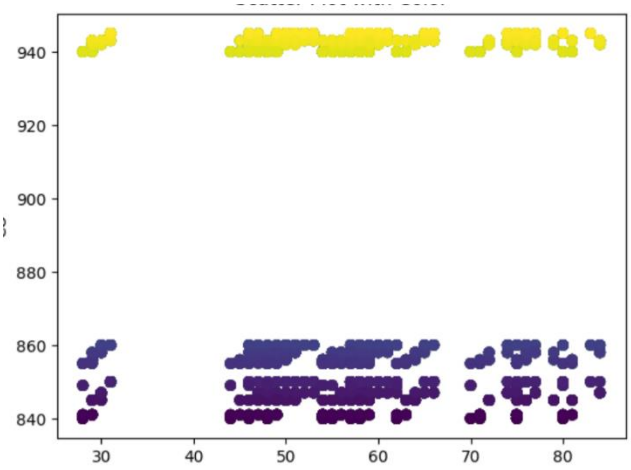


Fig 9: Scatter plot between all Parameters

Result represents the quantitative evaluation of deep learning model's performance

Evaluation Metrices

There are many metrices which allows us to evaluate models' accuracy and models performance. Metrices which are used in this project are

- **Mean Squared Error (MSE):** MSE is a

commonly used metric to evaluate the performance of regression and deep learning models. It measures the average squared difference between the predicted values by your model and the actual values of the predicted value. This also helps us to understand the loss of the model for better prediction.

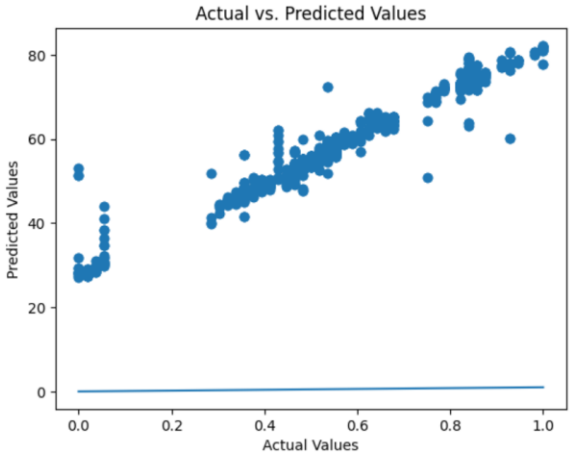


Fig 10: MSE Prediction Graph

MSE Formula

MSE = (1/n) * Σ (y_true_i - y_pred_i) ^2

Where,

- N = Number of data points
- y_true_i = actual target value for the i-th data point
- y_true_i = predicted value by the LSTM model for the i-th data point
- Σ (Sigma) = Summation of all data points

MSE helps us to understand the loss value of the value as input and predicted value. This will enhance the accuracy of the deep learning model. The more data undergoes iteration the more loss of the values decreases.

7. OUTPUT

DEEP LEARNING MODEL PREDICTED VALUES

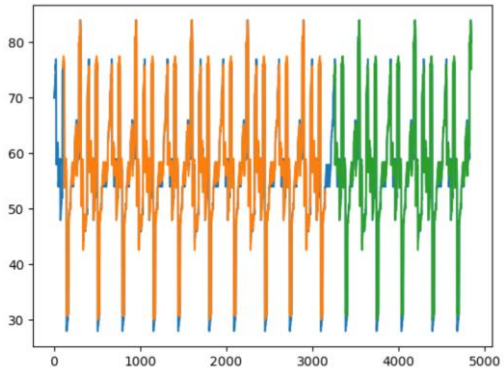


Fig 10: PM 2.5 Prediction

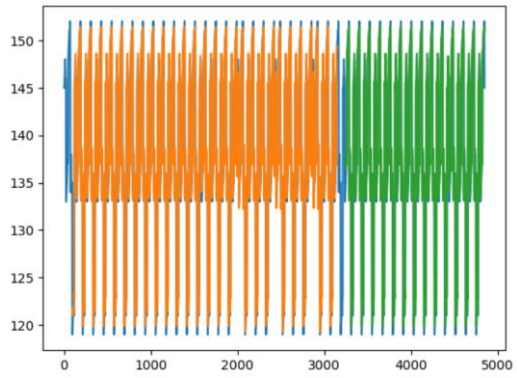


Fig 11: PM 10 Prediction

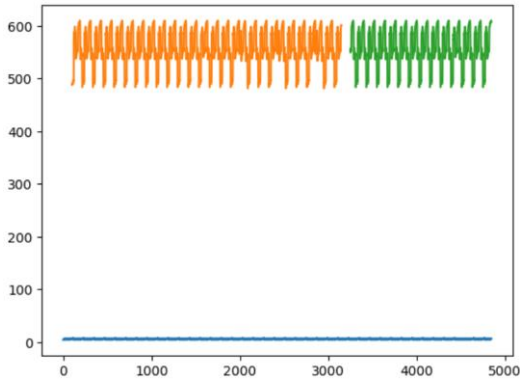


Fig 12: Ozone (O3) Prediction

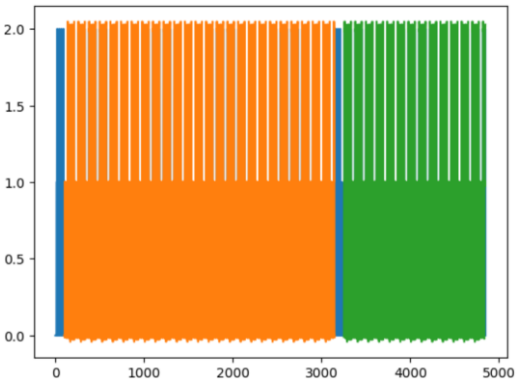


Fig 13: Sulphur Dioxide (SO2) Prediction

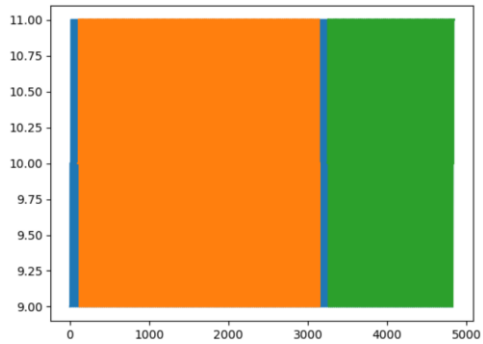


Fig 14: Nitrogen dioxide (NO2) Prediction

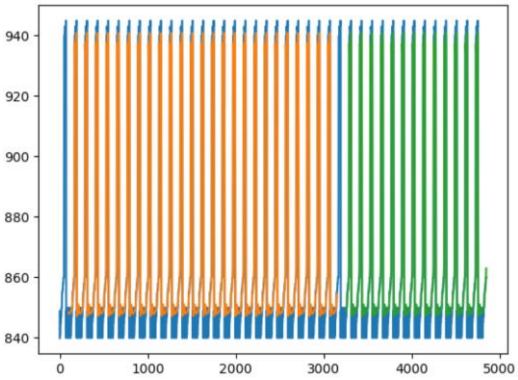


Fig 15: Carbon Monoxide (CO) Prediction

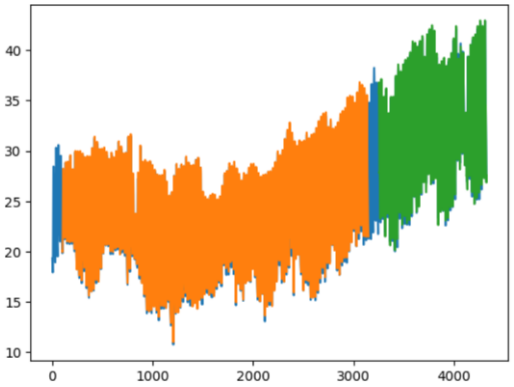


Fig 16: Temperature Prediction

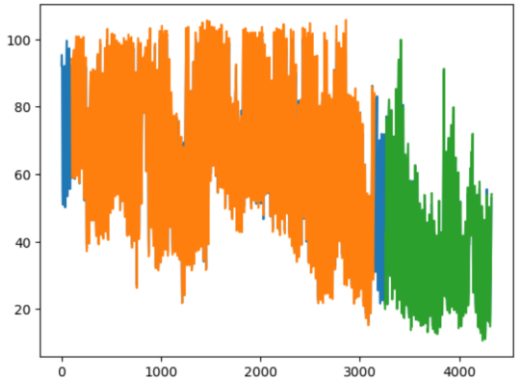


Fig 17: Humidity Prediction

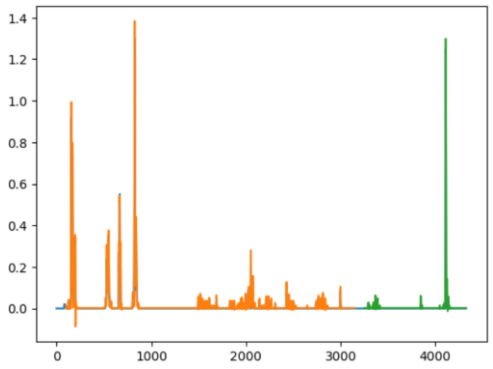


Fig 18: Precipitation Prediction

WEB APPLICATION LAUNCHING PAGE

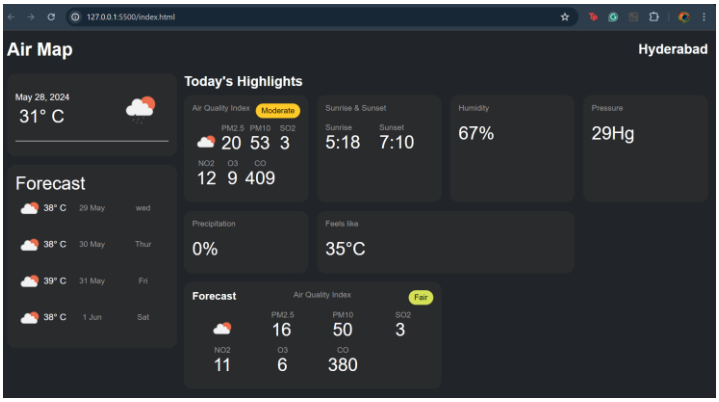


Fig 19: AIR MAP launching page

7. CONCLUSION

The project explored the potential of LSTMs for air quality forecast and was able to successfully model it. We showed how to use an LSTM model trained on historical air quality data, predicting trends in future air quality levels as it learns from complex patterns. This accomplishment is a stepping stone for future innovations in the area of air quality prediction system, which would be timely as well as accurate with defined threshold limits for Hyderabad. Finally, the development of Air Map is a crucial step towards helping with the air quality monitoring efforts. This interactive platform allows users to interact with the air quality data across locations and time periods.

The societal impact of the project is large The project will substantially raise awareness of air pollution in Hyderabad by establishing widespread availability and usability for any citizen with the results delivered as part of Air Map. Such awareness can enable citizens to demand better environmental protections, and it also ensures they monitor the steps taken by relevant

authorities to improve air quality. Improved decision-making on air quality is of great importance to people, especially those in vulnerable groups such as children, the elderly and individuals with respiratory diseases. Developers have allowed the users to customize their exploration by using Air Map with its incredible features.

REFERENCE

1.https://www.sciencedirect.com/science/article/pii/S0960982219310322 (Vol. 4, Detection of low concentration of air pollution, like cigarette smoke, cooking fumes, etc. is possible with the combination of an air quality sensor and data acquisition system. Ipp. 2188-2191). IEEE.

2.https://www.researchgate.net/publication/228807719_Air_Quality_Monitoring System *IEEE* transactions on vehicular technology, 54(3), pp.903-909.

3.https://sustainenvironres.biomedcentral.com primary goal of Sustainable Environment Research (SER) is to publish high quality research articles associated with sustainable environmental science and technology and to contribute to improving environmental practice. Moradi, K. and Nikmehr, S., 2012. *123*, pp.527-541.

4.https://www.sciencedirect.com/topics/earth-and-planetary-sciences/air-quality-monitoring. Air quality monitoring stations are placed at fixed sampling points that are representative for the different exposure situations in the various cities and regions.

5.https://www.sciencedirect.com/science/article/pii/S1110016824002485 A systematic survey of air quality prediction based on deep learning