Air Quality Index Monitoring and Prediction System

Shivam Rajawat, Sumeet Pavitrakar, Abnoor Singh, Kaif Saiyed
School of Computer Engineering and Technology
MIT-World Peace University, Pune, Maharashtra, India
Guide: Prof. Shilpa Sonawani
Email: shivam56rajawat@gmail.com,
sumeetbpavitrakar@gmail.com,
cheema.abnoor4@gmail.com,
kaifsaiyed896@gmail.com

Abstract - **Air quality has become a growing concern in recent years due to its impact on human health and the environment. To address this issue, a monitoring and prediction system has been developed to forecast air quality levels and provide real-time data on air quality conditions. This research paper explores the development and implementation of an Air Quality Index (AQI) monitoring and prediction system. The system utilizes machine learning algorithms to analyze historical and real-time data on various air pollutants to predict air quality levels. The system is designed to provide accurate and timely air quality information to the public. The AQI monitoring and prediction system can help individuals make informed decisions about their outdoor activities and reduce their exposure to air pollutants. It can also aid in the development of effective policies and regulations to improve air quality and protect public health.**

Overall, this research paper highlights the importance of monitoring and predicting air quality levels and how the AQI monitoring and prediction system can be a valuable tool in addressing air pollution. The system's effectiveness in providing accurate and reliable air quality information can contribute to a healthier and more sustainable environment for all.

Keywords – Air Quality Index; Time Series Prediction; SARIMA;ML;

I. INTRODUCTION

Air pollution is a growing concern worldwide due to its adverse impact on public health and the environment[1][4]. As a result, air quality monitoring and prediction systems have become crucial tools in the effort to reduce pollution levels and protect human health. In recent years, advancements in sensor technology and machine learning algorithms have made it possible to develop low-cost and effective air quality monitoring systems that can collect specific gas quantities in real-time. In this research paper, we aim to design a circuit that can collect data on specific gasses, such as carbon monoxide, Carbon Dioxide and PM2.5 dust particles [5]. and use this data to train a machine learning model to predict air quality levels. The circuit will consist of two gas sensor modules, a microcontroller unit, and SHARP dust sensor for data acquisition and processing. The collected data will be used to train a machine learning model that can predict air quality levels in real-time based on the measured gas quantities. We will evaluate the performance of the designed system by comparing its predictions with actual air quality data collected from official monitoring stations. This research aims to contribute to the development of low-cost and effective air quality monitoring systems that can help policymakers and citizens make informed decisions to mitigate pollution and improve air quality.

II. LITERATURE SURVEY

Research related to Air quality and its prediction have been increasing rapidly to deploy cost-effective solutions. There are many methods which have been used to get such results. Reference [6] developed an IoT based air quality monitoring system which consists of a low-cost sensor array, interfaced using I2C, controlled by using a HUZZAH32 development microcontroller board. This system is rechargeable and portable, and can also be added to a monitoring network. All the raw data and analysis is stored in the cloud and can be downloaded by the user, if required. These data are transmitted to the 3rd party service, which transmits the data to the smartphone app, in real-time, and displays it to the user. This model uses cloud to seamlessly transfer data to the network nodes which can directly be used on a user's application. In our case, we are uploading the data to Thingspeak which can then be directly sent to our Machine learning algorithm to be used in predicting the air quality.

Reference [22] , uses an integrated system consisting of a programmable microcontroller (Raspberry Pi 3), PPD42NS Dust Sensor and SIM800L GSM/GPRS module. This node is linked to a database that stores incoming data in real-time. The data received can then be displayed through an app, which uses Google Maps API and highlights the route in different colors for the user based on the Air Quality Index (AQI) of that particular area. They take readings from the PPD42NS Dust Sensor every 30 seconds. The average reading after observing for four minutes is sent to the database. To summarize, the authors of [22] have implemented a large-scale wide-area based pollution monitoring system where their main goal was to reduce the personal exposure to air pollution (both indoor & outdoor) of an individual by suggesting route maps with less pollution and warning the user of high levels of pollution. On the other hand, we are building a weather station that can detect the pollutant levels of a particular area and predict future pollution parameters accurately. Our readings will be taken once every hour and then sent to the thingspeak cloud. In Reference [23], the proposed system advances the Real-time Air Quality monitoring systems using ThingSpeak Cloud. Here, the system is implemented using GUI design along with Qt5. They implemented the same with Raspberry pi4 installed with the Raspbian operating system, Arduino Uno, CCS811 module interfaces and DHT11 interface which is used to transmit Temperature and Humidity information to Arduino Uno and also monitor in a cloud database. The gas sensor they used is Grove Air quality gas sensor V1.3. In our case, we are calibrating and combining gas sensors manually, whereas, in [23], they have used an already manufactured sensor system.

The authors of [24] use a Long Short Term Memory (LSTM) based air quality prediction system. They collected the air information from the end of 2013 to September 2018 from Shanghai,China. Our Prediction system will be based on using SARIMA (Seasonal Autoregressive Integrated Moving Average) as our model for prediction since it suits our data. We will also be using various other models such as ARIMA(Autoregressive Integrated Moving Average), SARIMA and Prophet to test their accuracy on our data.

III. RESEARCH METHODOLOGY

A. System Diagram

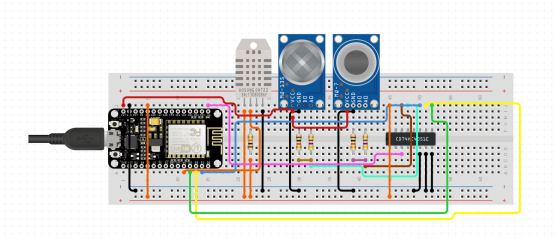


Fig1. Hardware Setup[3]

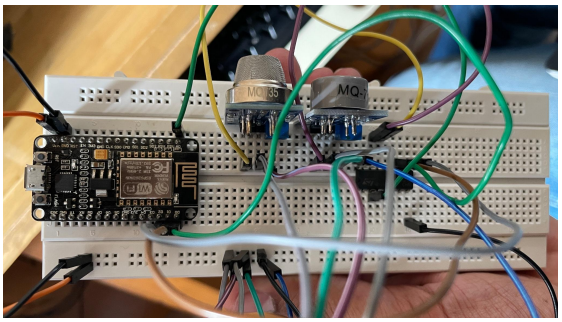


Fig2. Actual Implementation

Step1: Connect all the power pins (Vcc and GND) to the appropriate power source. In our case we are using the Vin pin on the ESP8266 to give power to the sensors.

Step2: Connect the analog outputs of the sensors to Input pins of MUX.

Step3: Connect the control pins of MUX to digital I/O of ESP8266. By setting the control pins to a particular number we will be able to read the Analog input from that input pin of MUX.

Step4: Connect the Output of MUX to Analog Input (A0) of the ESP8266.

Step5: Connect ESP8266 to Computer with micro USB cable and upload the code.

B. Block Diagram

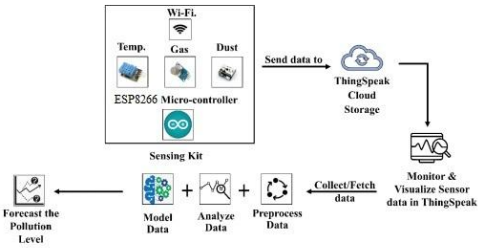
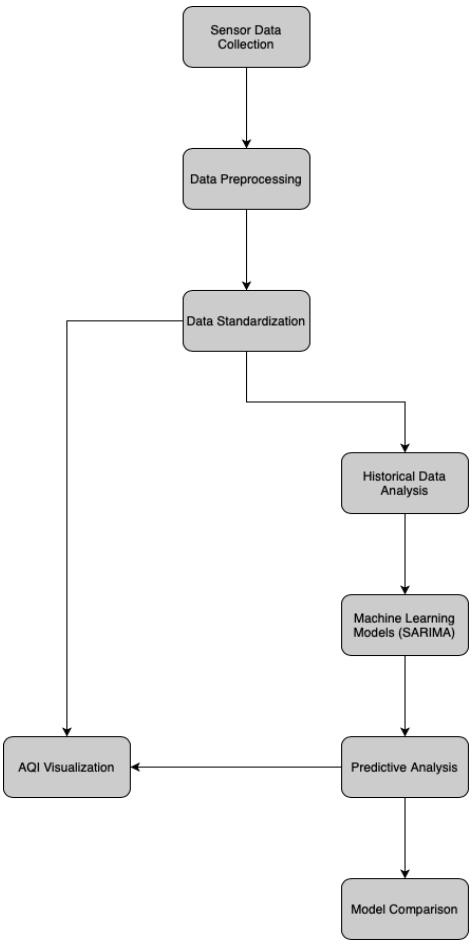


Fig3. Block Diagram[2]

Our System consists of three main parts:

1. Sensors that observe the data from the environment
2. Microprocessor (ESP8266) that performs calibration on the given data and establishes a link to the cloud by using on-board WiFi module.
3. Cloud Platform(ThingSpeak) which collects the data and allows us to Visually Analyze it and provides an API which can be used to forward the data for further processing.

C. Flowchart



1. Sensor Data Collection:
    - Air quality sensors are deployed at various locations to collect data on pollutants such as PM2.5,CO and CO2.

2. Data Preprocessing:
    - Raw sensor data is collected and transmitted to a central server or cloud platform.
    - Data is filtered, cleaned, and transformed to remove noise, outliers, and errors.
    - Data is standardized and converted to a common format and units for further analysis.

3. Data Standardization:
    - Sensor data is analyzed to calculate pollutant concentrations using standard formulas and algorithms.
    - Pollutant concentrations are mapped to the Air Quality Index (AQI) scale to obtain overall air quality readings.
    - AQI readings are stored in a database or time-series platform for real-time and historical analysis.

4. Historical Data Analysis:
    - Historical AQI data is analyzed to identify trends, patterns, and anomalies.
    - Statistical models and data mining techniques are used to extract insights and derive actionable recommendations.

5. Machine Learning Models:
    - Machine learning models like “SARIMA” are applied to predict future AQI values based on historical data, weather forecasts, and other factors.
    - Regression models, time series analysis are commonly used to forecast AQI values.

6. Predictive Analytics:
    - Predictive analytics algorithms are applied to forecast AQI values for the next hour.
    - Forecasts are updated in real-time based on new data and weather conditions.

7. AQI Visualization:
    - AQI readings are displayed on a dashboard or map for easy visualization.
    - Users can view AQI trends over time, compare AQI readings across different locations, and set up alerts for specific thresholds.

8. Model Comparison:
    - Comparing the accuracy of our model with MIT-WPU AQI Monitoring Station Database.

D. Model Explanation

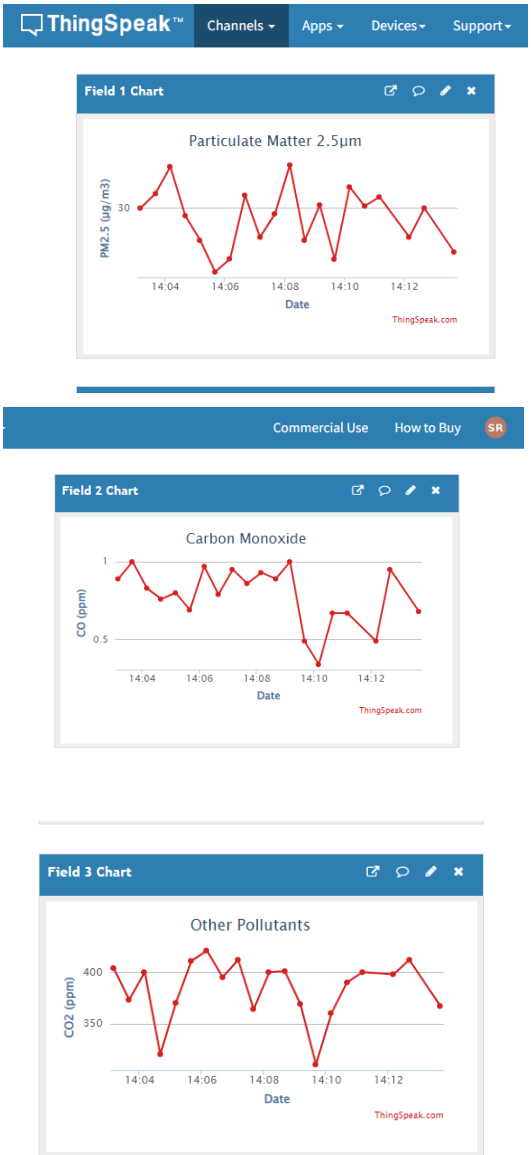
AQI (Air Quality Index) Monitoring and Prediction System is designed to measure air quality based on gasses like CO, CO2, Particulate Matter (PM2.5), Smoke. The air quality will be classified into three categories: Good, Moderate and Severe. The 'GP2Y1010AU0F PM2.5 Sharp Dust Sensor Module' is used to measure PM2.5 readings. The 'MQ-7 Gas Sensor' is used to measure Carbon Monoxide(CO) readings. The 'MQ-135 Gas Sensor' is used here to measure Carbon Monoxide and Smoke readings. The output of all these sensors is connected to the input pins of a multiplexer and the output pin of that multiplexer is connected to a 'NodeMCU ESP8266' Microcontroller. The multiplexer used is 'HCF4051BE'. The output of all sensors are sent to the microcontroller with a very small time gap of 100ms-500ms. This data will be uploaded to Thingspeak. After that, Machine Learning models are applied to the dataset collected to predict the value to air quality. This experiment will be compared to actual readings and it will also be compared with various renowned AQI platforms to check its accuracy. The ML model applied to the dataset is 'SARIMA'.

E. Hardware & Software Requirements

Hardware	**Model**	**Justification w.r.t project**
Particulate matter sensor	GP2Y1010AU0F PM2.5 Dust Sensor Module	To monitor wPM2.5 dust particles suspended in air.
Gas Sensor (CO)	MQ-7 Gas Sensor Module for Carbon Monoxide	To monitor Carbon Monoxide concentration in air.
Gas Sensor (smoke and other harmful gases)	MQ-135 Gas Sensor Module	To monitor Carbon Dioxide, Sulphide and Benze steam, also sensitive to smoke and other harmful gases.
Wifi Module, Microcontroller	ESP-12F ESP8266 Wifi Wireless IoT Board Module	Required for creating a gateway between sensors and cloud. Microcontroller required to connect all sensors and provide connection to the cloud.
Analog Multiplexer and Demultiplexer	74HC4051	Since ESP8266 only has one analog input so a Multiplexer is required to fetch each sensor value one by one

Software	Justification w.r.t project
Arduino IDE (C++)	Required to program the Arduino board to collect sensor data and forward it to the cloud.
Jupyter Notebook (Python)	Used to train the ML model.
Thingspeak	aggregate, visualize, and analyze live data streams in the cloud.

F. Monitoring the data through ThingSpeak



G. Selecting Model for Prediction

***a. Dataset* - MIT WPU Air Quality data from 6/5/2019 to 29/06/2022**

```
df.loc[:,['PM2.5 ug/m3','CO ppm', 'CO2 ppm']]
```

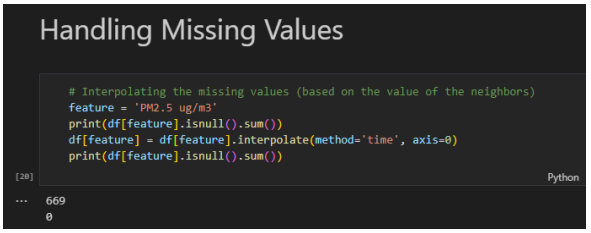
✓ 0.0s

timestamp	PM2.5 ug/m3	CO ppm	CO2 ppm
2019-06-05 01:00:00	29.95	NaN	NaN
2019-06-05 02:00:00	28.35	NaN	NaN
2019-06-05 03:00:00	32.55	NaN	NaN
2019-06-05 04:00:00	39.73	NaN	NaN
2019-06-05 05:00:00	33.58	NaN	NaN
...	...	...	...
2022-06-29 07:00:00	6.80	10.57	415.14
2022-06-29 08:00:00	7.30	10.03	413.11
2022-06-29 09:00:00	12.85	11.40	419.65
2022-06-29 10:00:00	8.75	11.40	426.25
2022-06-29 11:00:00	7.50	11.20	421.09

26891 rows × 3 columns

b. Data Preprocessing

Step1: Interpolating missing values

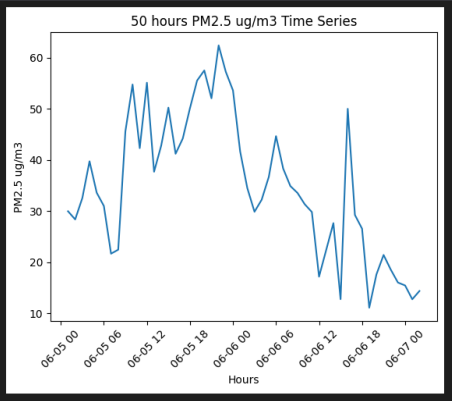


Step2:Time Series Plot

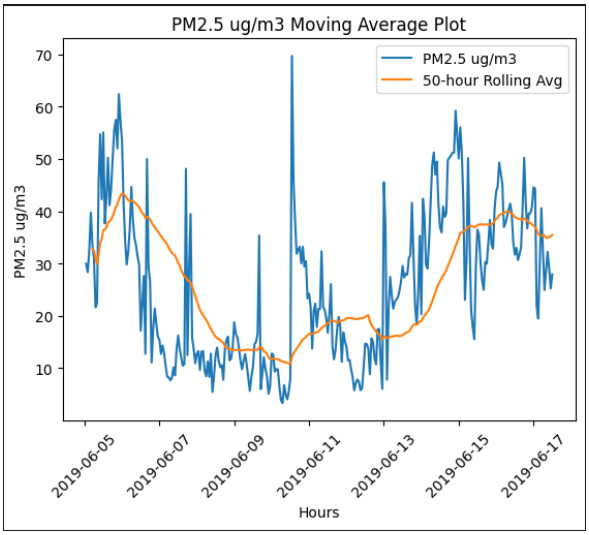
Time Series Plot

```
# Creating a time series plot using Matplotlib
hours = 50
plt.plot(df.index[0:hours], df[feature][0:hours])
plt.xlabel('Hours')
plt.ylabel(feature)
plt.title(str(hours) + ' hours '+ feature +' Time Series')
plt.xticks(rotation=45)
plt.show()
```

[21] Python

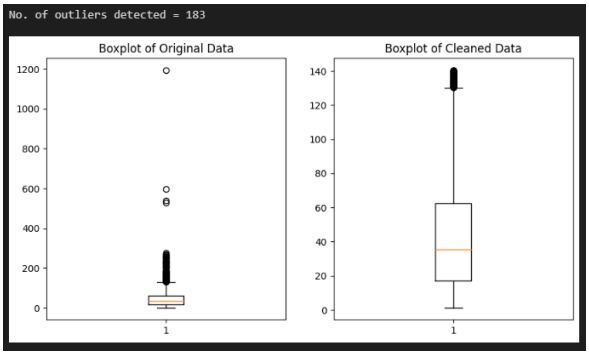


Step4: Moving Average Plot

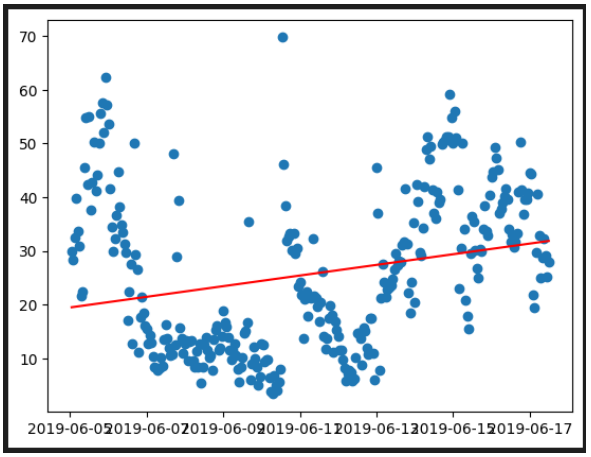


Step5: Handling the Outliers

```
No. of outliers detected = 183
```

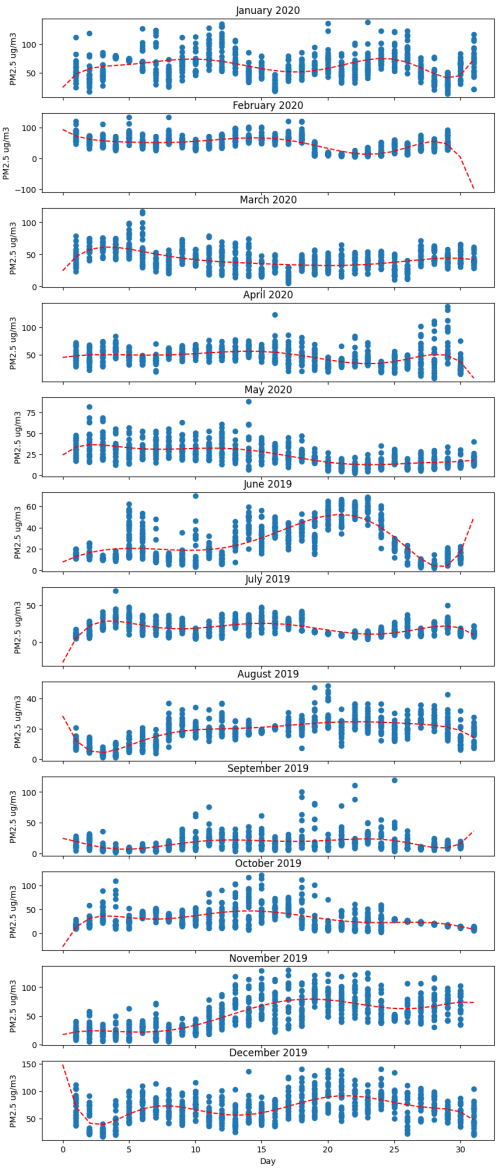


Step6: Linear Regression Plot

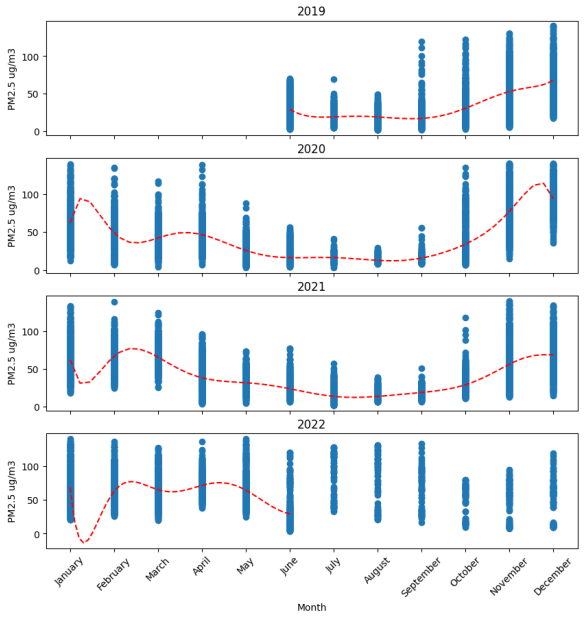


Step7: Seasonal Subseries Plot

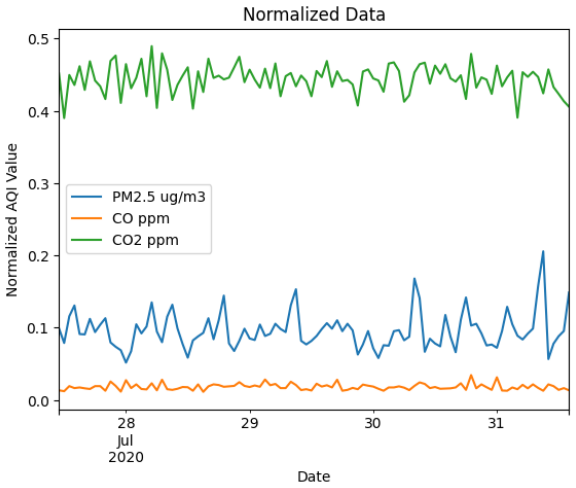
Monthly:



Yearly:



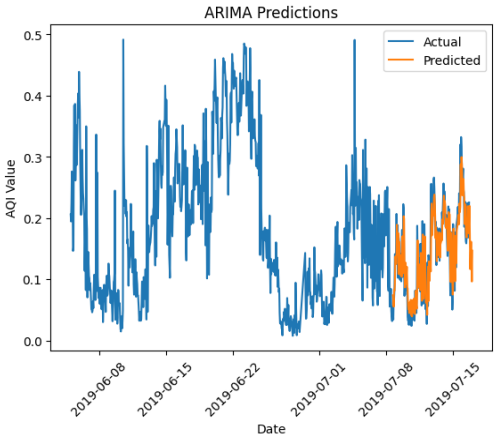
Step8: Normalization



c. Evaluating Best Model

1. ARIMA (AutoRegressive Integrated Moving Average)

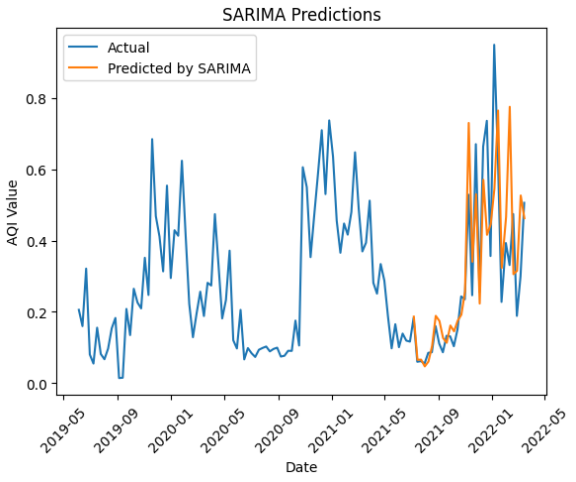
ARIMA models consist of three main components: the autoregressive (AR) component, the integrated (I) component, and the moving average (MA) component. The AR component involves using past values of the time series to predict future values. The MA component involves using past errors or residuals (the difference between the predicted and actual values) to make future predictions. The I component involves differencing the data to make it stationary and remove any trends or seasonality.[26]




```
Mean Absolute Error:  0.032856990582907294
Root Mean Squared Error:  0.046481436728349916
```

2. SARIMA (Seasonal Auto-regressive Moving Average)

SARIMA includes additional parameters to model the seasonal component of the data, making it better suited for time series with periodic patterns that repeat over a fixed interval of time. By incorporating these seasonal parameters, SARIMA can capture more nuanced seasonal patterns and provide more accurate forecasts than ARIMA.[28]

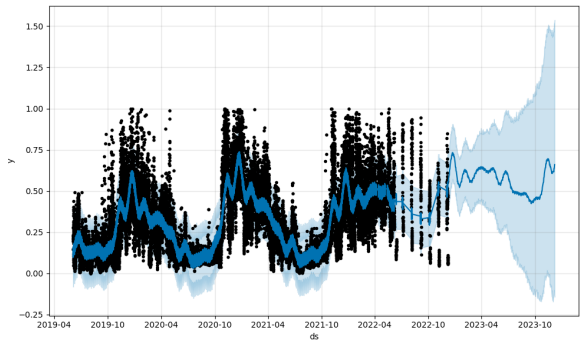



```
Mean Absolute Error:  0.012552720149467253
Root Mean Squared Error:  0.01724901171190559
```

3. Prophet

Prophet uses a decomposable time series model with three main components: trend, seasonality, and holidays. The trend component models the

overall direction of the time series, while the seasonality component captures periodic fluctuations. The holiday component allows users to specify known events or anomalies that may impact the time series.[27]




```
Mean Absolute Error:  0.3374483978058242
Root Mean Squared Error:  0.3913105413527864
```

Comparison:

Model	MAE	RMSE	TrainingTime
ARIMA	0.032	0.046	4.2s
SARIMA	0.012	0.017	15.4s
Prophet	0.337	0.391	36.2s

Platform: *Windows 10(Version 21H2), Python 3.9.4.*
Hardware: *Intel® Core i5(™) 9300H, 16GB DDR4 RAM.*

d. Selecting SARIMA as our Model

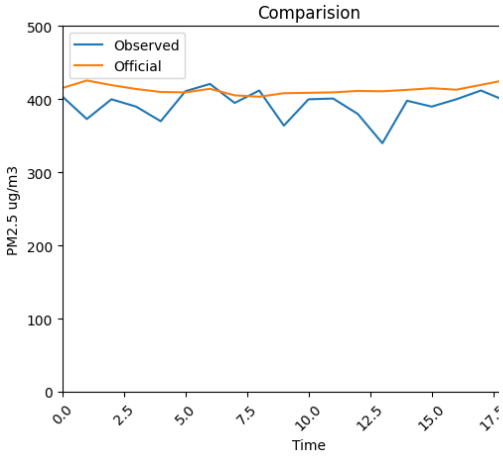
We noticed a Seasonality within our dataset which indicates that each year the same patterns repeat themselves and decided to go with SARIMA. As per our intuition it gives us the lowest MAE(Mean Absolute Error) and RMSE(Root Mean Squared Error) out of all algorithms though its training time was not the fastest but we can afford to make this tradeoff for higher accuracy. On the other hand the ARIMA model did not consider the seasonality of the data and the Prophet model tends to overfit the training data therefore giving less accurate results during testing.

IV. EXPERIMENTS & RESULTS

A. Observing Parameter & AQI Values from Our Sensors

timestamp	PM2.5 µg/m3	CO ppm	CO2 ppm
2023-05-01 14:03:10 IST	29.95	0.89	404
2023-05-01 14:03:40 IST	31.75	1	373
2023-05-01 14:04:10 IST	35.1	0.83	400
2023-05-01 14:04:40 IST	29	0.76	320
2023-05-01 14:05:10 IST	25.95	0.8	370
2023-05-01 14:05:40 IST	22.01	0.69	411
2023-05-01 14:06:10 IST	23.65	0.97	421
2023-05-01 14:06:40 IST	31.55	0.79	395
2023-05-01 14:07:10 IST	26.34	0.95	412
2023-05-01 14:07:40 IST	29.23	0.86	364
2023-05-01 14:08:10 IST	35.3	0.93	400
2023-05-01 14:08:40 IST	25.95	0.89	401
2023-05-01 14:09:10 IST	30.34	1	369
2023-05-01 14:09:40 IST	23.6	0.49	310
2023-05-01 14:10:10 IST	32.58	0.34	360
2023-05-01 14:10:40 IST	30.2	0.67	390
2023-05-01 14:11:10 IST	31.34	0.67	400
2023-05-01 14:12:40 IST	29.95	0.95	412
2023-05-01 14:12:10 IST	26.34	0.49	398
2023-05-01 14:13:40 IST	24.5	0.68	367

B. Comparing Observed data to predictions made by the model




```
variance = np.var(df[-20:]['CO2 ppm'] - observed_df[:]['CO2 ppm'])
print("Total Variance :",variance)
```

0.0s

```
Total Variance : 14.23
```

Variance was found to be 14.23, without normalization which is within the margin of error therefore we can say our model will suffice for further data prediction.

V. CONCLUSION

This paper describes the design and development of a Air Quality monitoring and prediction system, while using ThingSpeak for data gathering and SARIMA as the time series forecasting method. The data can be viewed in real time or reported as hourly or daily average. This implementation focuses on using low cost sensors and other equipment to monitor outdoor Air Quality. The total cost of the device was around $13, which included monitoring of PM2.5 dust particles, Carbon Monoxide (CO) and Carbon Dioxide (CO_2).

The main challenge faced during the making of this project was the collection of data and preprocessing the dataset so that it can be used to train the SARIMA ML model.

Future Research in the field of machine learning can work upon decreasing the Mean Absolute Error or finding a better prediction algorithm such as Temporal Fusion Transformer(TFT), on the other hand IoT based research can improve upon sensor calibration and overall choice of sensors for better real world view of air quality.

VI. REFERENCES

[1] World Health Organization, 2014. Ambient (outdoor) air pollution in cities database 2014. *World Health Organization, Retrieved from:⟨ http://www. who. int/phe/health_topics/outdoorair/databases/cities /en*.

[2] Fig1. ,M. Rakib, S. Haq, M. I. Hossain and T. Rahman, "IoT Based Air Pollution Monitoring & Prediction System," 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 2022, pp. 184-189, doi: 10.1109/ICISET54810.2022.9775871.

[3] Circuito.io

[4] Laumbach RJ, Kipen HM. Respiratory health effects of air pollution: update on biomass smoke and traffic pollution. J Allergy Clin Immunol. 2012;129(1):3–11. Crossref, Medline, Google Scholar

[5] Centers for Disease Control and Prevention, “Air Pollutants | Air - CDC” [Air Pollutants]

[6] S. Esfahani, P. Rollins, J. P. Specht, M. Cole and J. W. Gardner, "Smart City Battery Operated IoT Based Indoor Air Quality Monitoring System," 2020 IEEE SENSORS, Rotterdam, Netherlands, 2020, pp. 1-4, doi: 10.1109/SENSORS47125.2020.9278913.

[7] https://www.waveshare.com/wiki/MQ-7_Gas_Sensor

[8] https://www.pololu.com/file/0J313/MQ7.pdf

[9] https://electronicsprojects.in/wp-content/uploads/2022/11/MQ7-Gas-Sensor-Pin-Diagram-Image.png

[10]
https://adiy.in/shop/sensors/gas-sensors/mq135-air-quality-sensor-module/

[11]
https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ135%20(Ver1.4)%20-%20Manual.pdf

[12]
https://adiy.in/shop/sensors/gas-sensors/mq135-air-quality-sensor-module/

[13]
https://www.electronicscomp.com/gp2y1010au0f-optical-dust-sensor-module?gclid=CjwKCAjwl6OiBhA2EiwAuUwWZes6bwZdQBUu-qWMwl0Zo8yecazqTN-7EsCEgVlKP-qqEq71GLCJAhoCFs8QAvD_BwE

[14]
https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf

[15]
https://microcontrollerslab.com/gp2y1014au0f-dust-sensor-pinout-working-applications-datasheet/

[16]
https://www.amazon.in/Easy-Electronics-NodeMcu-Development-Board/dp/B06XYRS6KC

[17]
https://www.make-it.ca/nodemcu-details-specifications/

[18]
https://esp8266-shop.com/esp8266-guide/esp8266-nodemcu-pinout/

[19]
https://www.amazon.in/Invento-2pcs-HCF4051-4051-Demultiplexer/dp/B0746H5DC7

[20]
https://www.st.com/en/switches-and-multiplexers/hcf4051.html

[21]
https://www.soemtron.org/downloads/disposals/hcf4051be.pdf

[22] Pullan, Pearl & gautam, chitra & Niranjan, Vandana. (2020). Air Quality Management System. 10.1109/GUCON48875.2020.9231233.

[23] S. Faiazuddin, M. V. Lakshmaiah, K. T. Alam and M. Ravikiran, "IoT based Indoor Air Quality Monitoring system using Raspberry Pi4," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2020, pp. 714-719, doi: 10.1109/ICECA49313.2020.9297442.

[24] Y. Jiao, Z. Wang and Y. Zhang, "Prediction of Air Quality Index Based on LSTM," *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2019, pp. 17-20, doi: 10.1109/ITAIC.2019.8785602.

[25] M. Rakib, S. Haq, M. I. Hossain and T. Rahman, "IoT Based Air Pollution Monitoring & Prediction System," *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Chittagong, Bangladesh, 2022, pp. 184-189, doi: 10.1109/ICISET54810.2022.9775871.

[26]
https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#toc-what-is-an-autoregressive-integrated-moving-average-arima

[27]
https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/

[28]

https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

[29]
https://github.com/esp8266/esp8266.github.io/tree/master/stable

[30]
https://github.com/sid56rajawat/AQI-Monitoring-and-Prediction