Dissertation on

Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS461A – Capstone Project Phase - 1

Submitted by:

Aditi Jain	**PES2UG20CS021**
Aditya R Shenoy	**PES2UG20CS025**
Ananya Adiga	**PES2UG20CS043**
Anirudha Anekal	**PES2UG20CS051**

Under the guidance of

Prof. Saritha
Assistant Professor
PES University

June - Nov 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

is a bonafide work carried out by

Aditi Jain	**PES2UG20CS021**
Aditya R Shenoy	**PES2UG20CS025**
Ananya Adiga	**PES2UG20CS043**
Anirudha Anekal	**PES2UG20CS051**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE20CS461A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2023 – Nov. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature **Prof. Saritha** Assistant Professor	Signature Dr. Sandesh B J Chairperson	Signature Dr. B K Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. Dr. Sarasvathi V	_____
2. Prof.Komal Baheti	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **Monitoring the concentration of air pollutants and its health hazards using Machine Learning models** has been carried out by us under the guidance of Prof. Saritha, Assistant Professor, PES University and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2023. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

Name	**SRN**	**Signature**
Aditi Jain	PES2UG20CS021	
Aditya R Shenoy	PES2UG20CS025	
Ananya Adiga	PES2UG20CS043	
Anirudha Anekal	PES2UG20CS051	

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Saritha, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE20CS461A - Capstone Project Phase – 2.

I am grateful to the Capstone Project Coordinator, Dr. Sarasvathi V, Professor and Dr. Sudeepa Roy Dey, Associate Professor, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University and Prof. Nagarjuna Sadineni, Pro-Vice Chancellor - PES University, for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

In our modern era, industrialization poses a significant threat to air quality, with pollutants such as PM2.5, PM10, and CO adversely affecting human health, particularly the respiratory system. Widespread ignorance compounds this issue, necessitating urgent awareness campaigns to inform individuals about the omnipresent dangers they face with each breath.

Our proposed system takes on the crucial role of continuously monitoring air quality in the user's surroundings, issuing alerts about potential risks, specifically the threat of developing lung cancer. This research-driven initiative seeks to improve existing systems by enhancing adaptability to contemporary needs and efficiently managing the influx of data. To achieve this, we employ a hybrid model that integrates Adaptive LSTM and ARIMA models, known for their effectiveness in handling time series data. These models are trained to adapt to dynamic changes, ensuring reliability in the face of evolving environmental conditions.

After the training phase, our model is deployed on a secure cloud platform to ensure scalability and accessibility for users, aligning it with current demands. Real-time air quality data is collected using Arduino boards and high-quality sensors deployed in specific environments. This data informs predictions, contributing to the continuous learning of the model and refining its accuracy over time. Our project aims to raise public awareness, improve predictive accuracy, and, ultimately, contribute to saving lives while fostering a sense of environmental responsibility in both governmental and industrial sectors for the benefit of future generations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I
INTRODUCTION

1.1 Motivation

The industrialization process has been a significant contributor to the current pollution crisis facing the world. However, many people remain unaware of the dangers associated with poor air quality. PM2.5, PM10, NO2, SO2, and CO are examples of pollutants which are particularly harmful to human health, with the lungs being the first point of contact. Particulate matter of a small diameter enters the lungs easily. As such, it is crucial to increase public awareness of these hazards and encourage individuals to take protective measures.

To address this issue, it is necessary to develop and implement air quality monitoring systems. Such systems would provide individuals with real-time information on air quality levels in their surroundings and alert them to potential health risks. By increasing awareness and providing early warnings, that can help individuals take preventative measures and protect their health.

1.2 Solution

Therefore, the proposed system is a continuous air quality monitoring system that will keep track of the air quality in the user's vicinity and alert them to the likelihood of developing lung cancer. This research-based project aims to improve the effectiveness of existing systems to ensure maximum efficiency, taking into account the current volume of data.

The proposed system is designed to monitor the levels of PM2.5, PM10 and CO in the air continuously, providing the user with real-time information about the quality of the air they are breathing. The system will then use this data to assess the user's risk of developing lung cancer and

alert them accordingly. By keeping track of the air quality and providing early warnings, this system can help people take appropriate measures to safeguard their health.

In conclusion, the proposed air quality monitoring system is an essential tool for ensuring people's health and well-being in today's polluted environment. With its ability to continuously monitor the air quality and provide real-time alerts, this system will enable people to take preventive measures to protect their health from the harmful effects of air pollution.

CHAPTER II
PROBLEM STATEMENT

2.1 Description

To address the issue of increasing mortality rate due to air pollution , the model proposed is a hybrid model of Adaptive LSTM and ARIMA deep learning models that can predict the probability of a person being affected by the air quality in his surroundings.

The model will be deployed on a cloud platform which ensures scalability and accessibility to store huge amounts of real time data of air quality collected using arduino boards and high quality IOT sensors. Continuously monitoring air quality, predicting and alerting individuals to potential health risks. These deep learning models have proven to be effective in handling time series data and are easy to implement. Moreover, they have high reliability and can dynamically adjust to changes, making them ideal for real-time monitoring of air quality.

By constantly training on real-time and static data, the model will become increasingly accurate, allowing it to provide more reliable and relevant predictions. This will enable individuals to take preventive measures to protect themselves from the harmful effects of air pollution and make informed decisions about their daily activities.

CHAPTER III
LITERATURE REVIEW

3.1 Determination of Air Quality Life Index (AQLI) in Medinipur City of West Bengal(India) During 2019 To 2020 : A contextual Study

3.1.1 Approach & Results:

- They do not use any ML model. They use the predefined method provided by various organizations like WHO, INAAQS etc.
- They show a provable reduction in life expectancy and the correlation to various air pollutants.
- They also speak about the direct effects of PM2.5 on the lungs.

3.1.2 Advantages:

- They use an air pollutant rather than AQI.
- Speaks about the direct effects of PM on various parts of the body.
- Uses an expensive but very accurate, and lab tested/ approved sensor for PM 2.5
- (Plantower particulate matter sensor PMS3003)

3.1.3 Limitations:

- No ML model used. They use existing methods to determine everything.
- Only looks at one pollutant.
- Not clear if all the effects (of the pollutant) are considered .

3.2 The nexus between COVID-19 deaths, air pollution and economic growth in New York state: Evidence from Deep Machine Learning.

3.2.1 Approach & Results:

- They use an ensemble of ML models (ANN, Deep Learning, Decision Tree)

- The association between COVID-19-related mortality, economic growth, PM10, PM2.5, and NO2 concentrations in New York state is examined in this article.

3.2.2 Advantages:

- They consider individual pollutants.
- Very well thought out data gathering and cleaning.
- The use methods used are well suited for the type of predictions and the type of data being used.

3.2.3 Limitations:

- It has been done mainly to covid related deaths.
- This has many inherent issues especially considering how little we truly know about covid.
- They attribute over 50% of deaths due to covid as deaths facilitated by these pollutants.
- This cannot be proven due to the lack of research time on the effects of covid.
- Does not consider other underlying issues

3.3 An Application of IoT and Machine Learning to Air Pollution Monitoring in Smart Cities

3.3.1 Approach & Results:

- They use ANN models for predicting the concentrations of air contaminants and Pearson's coefficient for the correlation with weather conditions.
- The objective of this paper is to monitor pollutant concentrations in smart cities and find a correlation between pollutants and weather parameters and prediction of the pollutants to prevent diseases like lung cancer.

3.3.2 Advantages:

- The paper takes into account the weather conditions as well. As air pollutants mixed with other factors like water/high winds cause differing effects
- The model created has achieved a Root Mean Square Error of only 0.0128 for SO2 prediction and 0.0001 for PM2.5

- They have used two different methods (Pearson Correlation and ANN) for the weather correlation and prediction.

3.3.3 Limitations:

- Does not consider all the environmental weather conditions (Only wind speeds, temperature and humidity)
- The model only contains a single hidden layer and they have not run it only for 500 epochs
- The IoT infrastructure they have used is very crowded and complex.

3.4 Mohammad Marzouk and Mohammad Atef "Assessment of indoor air quality in academic buildings using IOT and deep learnings"

3.4.1 Approach & Results:

To find the correlation between outdoor pollution and indoor air quality, by monitoring real time IAQ using IOT sensors and sending the collected data to the cloud via wireless connections.CNN and LSTM are used to train the collected data.

3.4.2 Advantages:

- Uses real time data.
- Finds the correlation between outdoor and indoor environments.
- Concentrates on rural areas too, which might be affected by burning wood, construction particles and unpaved roads.

3.4.3 Limitations:

- Does not consider the family history with lung cancers.
- Does not consider the occupational exposures to carcinogens.
- It is biased as the readings were collected only during summer when the humidity will be relatively high.
- The study was restricted to just a few primary pollutants and did not consider volatile organic compounds, NH3 and O3

3.5 Jin Z-Y, Wu M, Han R-Q, Zhang X-F, Wang X-S, et al. (2014) "Household Ventilation May Reduce Effects of Indoor Air Pollutants for Prevention of Lung Cancer: A Case-Control Study in a Chinese Population.

3.5.1 Approach & Results:

The purpose of this research is to investigate the link between home ventilation and lung cancer.
A standardized questionnaire was used to obtain epidemiologic and home ventilation data.
The adjusted odds ratios (ORadj) and 95% confidence intervals (CI) were calculated using unconstrained logistic regression.

3.5.2 Advantages:

- The information gathered takes into account a person's family history of lung cancer, basic demographic characteristics, socioeconomic status, tobacco smoking history, alcohol intake, dietary history, and physical activity.
- Active smokers, second hand smokers, carcinogens, tobacco smoking and high temperature cooking oil flames were considered for the study thereby covering a wide scope.

3.5.3 Limitations:

- Because the data is limited to the local community, selection and memory bias may present in the study.
- The data about family history, age, gender is considered via a standardized questionnaire by interviewers, using quantitative data might have been effective instead.
- Have not considered occupational exposure
- The study is limited to IAQ and does not consider how outdoor pollutants are affecting IAQ.

3.6 How Is the Lung Cancer Incidence Rate Associated with Environmental Risks? Machine-Learning-Based Modeling and Benchmarking

3.6.1 Approach & Results:

- They use an ensemble of models like Logistic Regression, Random Forest, SVM and Gradient Boosting and perform benchmarking on them.
- The goal of this research is to use machine-learning-based modeling and benchmarking to examine the association between lung cancer incidence rate and air pollution.
- The study's goal is to create a prediction model to better understand the influence of air pollution on illness.

3.6.2 Advantages:

- They perform benchmarking, i.e, comparing the performance of different ML models and provides insights on which the best one is .
- The dataset they have used contains both environmental risk factors (air pollutants) and the lung cancer incidence rates from various countries and hence increases generalizability.

3.6.3 Limitations:

- Does not consider all the environmental weather conditions (Only wind speeds, temperature and humidity)
- The model only contains a single hidden layer and they have not run it only for 500 epochs
- The IoT infrastructure they have used is very crowded and complex.

3.7 Kim Kyung Eun, Cho Daeho, Park Hyun Jeong, Air pollution and skin diseases: Adverse effects of airborne particulate matter on various skin diseases, Life Sciences

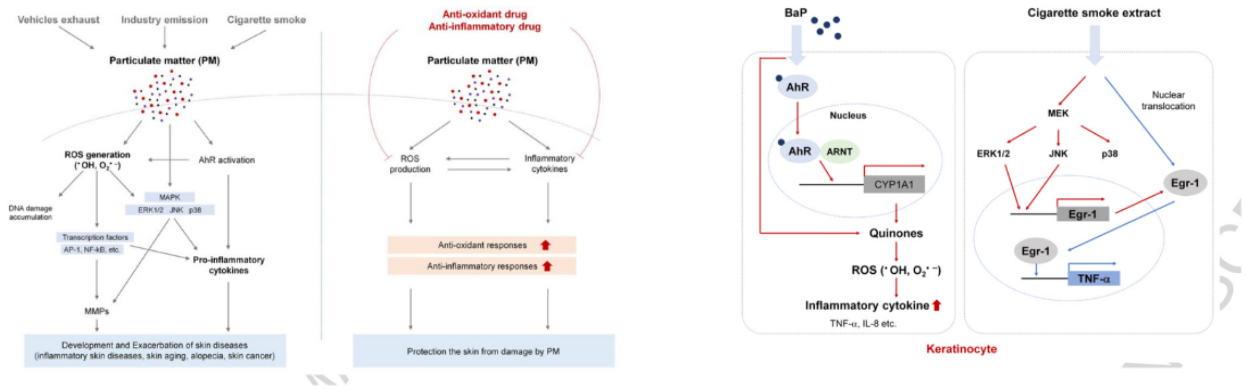

Fig 3.7.1- Retention of pollutants into the skin

3.7.1 Approach & Results:

- This page discusses the relationship between PM and skin illnesses, as well as the underlying immunological pathways.
- Increased PM levels have been linked to the development of a variety of skin illnesses through the control of oxidative stress and inflammatory cytokines.
- As a result, antioxidant and anti-inflammatory medicines may be beneficial in the treatment of PMS-induced skin disorders.

3.7.2 Advantages:

- This paper has listed in detail which air pollutants are known to cause which disease.
- It lists various diseases like Atopic dermatitis, Acne, Psoriasis, and skin cancer.
- It also discusses the conditions of skin aging, alopecia, and oxidative stress.

3.7.3 Limitations:

- The amount of statistics is less than is used to prove its point.
- It doesn't discuss the correlation between these diseases, and how they affect the probability of another happening.

3.8 Roberto Cazzolla Gatti, Arianna Di Paola, Alfonso Monaco, Alena Velichevskaya, Nicola Amoroso, Roberto Bellotti, The spatial association between environmental pollution and long-term cancer mortality in Italy, Science of The Total Environment,

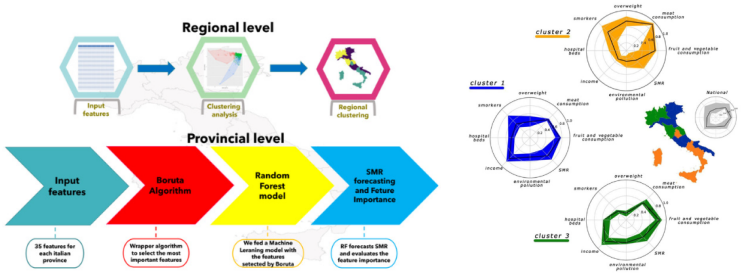

Fig 3.8.1:- Methodology and cluster analysis output

3.8.1 Approach & Results:

- Using an artificial intelligence method, this article examined the relationships between cancer mortality, socioeconomic characteristics, and sources of environmental pollution in Italy at both broad regional and narrower provincial scales.
- Random Forest (RF) regression in combination with a Boruta feature importance analysis, K means clustering.
- SMR forecasting and Feature importance, Regional cluster analysis.

3.8.2 Advantages:

- It has taken into consideration all the different types of body parts that can get affected due to air pollution in extensive detail.
- The data sources and the algorithms used have been discussed.
- Investigated the possible geographical relationship between socioeconomic and lifestyle characteristics.

3.8.3 Limitations:

- Some individual sources of pollution have no link to any form of cancer. This study is limited to Italy, but it should be expanded to other nations as well, because air pollution levels vary by region.

3.9 Conclusion to the literature survey:

Most studies only include a small number of risk variables when examining the influence of air quality on human health. Some initiatives, for example, consider just indoor air quality (IAQ) or outdoor air quality (OAQ), but not both, or they exclude environmental variables, family history, and occupational exposure. This method can lead to biased and untrustworthy results that do not completely reflect the complexities of the topic at hand.

Furthermore, despite the fact that air pollution is a major public health problem, little study has been conducted on its real health impacts. While numerous research have looked at the association between air quality and health outcomes, few have looked at the causative relationship. Due to a dearth of study, properly predicting the influence of air quality on health can be difficult.

Another problem is that projections based on static and out-of-date data are uncommon. This can be an issue since air quality can fluctuate fast owing to a number of variables such as weather, industrial activity, and transportation. Predictions based on old data might result in incorrect conclusions and perhaps dangerous choices.

To address these difficulties, more extensive and sophisticated models that reflect a larger variety of risk variables and the real health impacts of air pollution are required. These models must also be based on real-time data and capable of fast adapting to changes in air quality levels. This allows us to create more precise forecasts regarding the impact of air pollution on human health, which may guide policy decisions and assist individuals in making educated decisions about their health and well-being.

CHAPTER IV
DATA

4.1 Overview

The aim is to use a static dataset containing correlation between lung cancer and air pollutants, for training the model. Two datasets have been used for solving this approach.

The lung cancer patient dataset has been acquired from data.world which is a data catalog platform built to provide open access to generative AI ready datasets.

Link to dataset: https://data.world/cancerdatahp/lung-cancer-data

The Air Quality Data in India (2015 - 2020) dataset which has hourly data across stations and cities in India has been taken from kaggle.com, where it had been collected and compiled from publicly available data given by the Central Pollution Control Board, an official portal by the Govt. of India.

Link to dataset: https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india

4.2 AQI dataset

	StationId	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	AP001	2017-11-24	71.36	115.75	1.75	20.65	12.40	12.19	0.10	10.76	109.26	0.17	5.92	0.10	NaN	NaN
1	AP001	2017-11-25	81.40	124.50	1.44	20.50	12.08	10.72	0.12	15.24	127.09	0.20	6.50	0.06	184.0	Moderate
2	AP001	2017-11-26	78.32	129.06	1.26	26.00	14.85	10.28	0.14	26.96	117.44	0.22	7.95	0.08	197.0	Moderate
3	AP001	2017-11-27	88.76	135.32	6.60	30.85	21.77	12.91	0.11	33.59	111.81	0.29	7.63	0.12	198.0	Moderate
4	AP001	2017-11-28	64.18	104.09	2.56	28.07	17.01	11.42	0.09	19.00	138.18	0.17	5.02	0.07	188.0	Moderate

Fig 4.2.1: AQI dataset description

The dataset contains 108035 rows and 16 columns.

Table 4.2.1- The below table mentions the attributes in the dataset and its description.-

Attributes	**Attribute description**
StationId	Id of station
Date	Date the record was recorded on
PM 2.5	Particulate Matter 2.5-micrometer in ug / m3
PM 10	Particulate Matter 10-micrometer in ug / m3
NO	Nitric Oxide in ug / m3
NO2	Nitric Dioxide in ug / m3
NOx	Any Nitric x-oxide in ppb
NH3	Ammonia in ug / m3
CO	Carbon Monoxide in ug / m3
SO2	Sulphur Dioxide in ug / m3

O3	Ozone in ug / m3
Benzene	Benzene in ug / m3
Toluene	Toluene in ug / m3
Xylene	Xylene in ug / m3
AQI	Air Quality Index
AQI_Bucket	Air Quality Index bucket

.

4.3 Lung cancer dataset

	Patient Id	Age	Gender	AQI	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Smoking	Passive Smoker	Clubbing of Finger Nails	Frequent Cold	Level
0	P1	33	1	2	5	4	3	2	3	2	1	2	Low
1	P10	17	1	3	5	3	4	2	2	4	2	1	Medium
2	P100	35	1	4	6	5	5	4	2	3	4	6	High
3	P1000	37	1	7	7	7	6	7	7	7	5	6	High
4	P101	46	1	6	7	7	7	6	8	7	2	4	High

Fig 4.3.1: Lung cancer dataset description

The dataset contains 1001 rows and 25 columns.

We have only used 12 columns for our project as they are the ones that are valid for our research.

Columns such as Age, Gender, AQI, Dust Allergy, Occupational hazards, etc. have been used.

Columns like Alcohol use, Obesity, etc. have been dropped.

CHAPTER V
PROJECT REQUIREMENTS SPECIFICATION

5.1 Introduction

Proposed is a hybrid Adaptive LSTM and ARIMA model for continuous air quality monitoring in response to growing concerns about air pollution. Deployed on a secure cloud platform for scalability, the model utilizes real-time data from Arduino boards and sensors to enhance predictions. The goal is to improve accuracy, raise awareness, and promote environmentally conscious decisions to mitigate health risks.

5.2 System Requirements

5.2.1 Features

The product features entail:

- The user will be able to check the current levels of pollution in his surroundings and the probability of contracting lung cancer based on the current exposure to pollutants.
- The user can check the potential probability of contracting lung cancer. This will be derived from results from the machine learning model.

5.2.2 User classes and characteristics

- **Smart home companies:**

  Smart home companies currently incorporate various sensors like temperature sensors, heat sensors and motion sensors. They can also incorporate the pollutant sensors as an additional layer of detection on the current infrastructure.

- **Government:**

  Governments can use our product to identify pollution hotspots and create policies and regulations to reduce the harmful emissions and issue public health warnings in high risk areas.

- **City planning:**

  City planners and private companies can use this information to guide urban planning and development and create communities in a healthier living space.

- **General Public:**

  The general should be informed about the levels of pollution and the risks associated with it to take preventive measures

5.2.3 Operating Environment

- **Arduino:**

  The default operating system present on the board will be used. A custom code will be added to it. Any OS that supports the Arduino IDE (or the web version) can be used to interface with the Arduino (this is only to add code to the board).

- **Coding Platform:**

  Any OS and any editor can be used that you prefer as long as it supports git or some other way to have a shared codebase. It should also have inbuilt support for languages that will be used for coding and a way to integrate databases.

5.2.4 Software requirements

- **Data collection and preprocessing tools**:

  Tools to collect air pollutant data from various sources, such as air quality sensors and software to clean, normalize, and prepare the data for analysis will be used.

- **Machine Learning frameworks:**

  Machine learning frameworks such as TensorFlow, Keras, will be used to build and train the models. These frameworks provide a range of tools for data processing, model building, and evaluation.

- **Statistical analysis tools:**

  Statistical analysis tools such as Python packages like NumPy ,Pandas and Statsmodels will be used for data analysis and exploratory data analysis (EDA).

- **Visualization tools:**

  The project will need visualization tools like Matplotlib, Seaborn to create visualizations and graphs that help you understand the data and communicate results.

- **Cloud computing platforms:** To store and analyse massive amounts of data and deploy our machine learning models, the project will also utilize a cloud computing platform such as Streamlit and Thingspeak.

5.2.5 Hardware requirements

- ESP8266 wifi module to collect data from sensors and store it in the ThingSpeak cloud platform.

- MQ7 to monitor the concentration of CO in the atmosphere.
- GP2Y1010AUF Dust sensor to monitor particulate matter of different aerodynamic diameter (PM2.5 and PM10)

5.2.6 User Interface

- Basic information can be displayed on an LCD connected to the stations. More data can be viewed on a computer with basic internet connection (either through a custom made website/ application or the actual cloud platform directly).
- In the provided interface there will be basic instructions on how to interpret the data received.
- The stations can measure pollutants every 20 seconds. It can send each measurement to the database, preferably immediately. The model can rerun with the new received data every time the user requests.
- All error messages will be sent to the cloud platform. Some errors can be displayed on the LCD.

5.2.7 Communication interface

- **Arduino:**

  The default operating system present on the board will be used. The project will use custom code written. Any OS that supports the arduino IDE (or the web version) can be used to interface with the arduino (this is only to add code to the board) .
- **Cloud:**

The project will use a few cloud platforms for this. This will evolve as time progresses and pros and cons are found out for each platform. Thinkspeak to save the data from the sensor stations, AWS Glue, Google Cloud Dataflow or Streamlit to process the data and to host the model.

5.3 Functional Requirements

- **Machine Learning Algorithms:** Implement ML algorithms to analyze the air quality data and find the correlations between pollutants and lung cancer rates.
- **Cloud Infrastructure:** Use a cloud infrastructure such as Streamlit, AWS, Google Cloud Platform, or Microsoft Azure to host the ML models and the database.
- **Deployment:** Deploy the ML models to the cloud infrastructure and provide a user friendly interface to allow users to access the models and the data.
- **Real-time Monitoring:** Continuously monitor the air quality using IoT devices and update the database in real-time. The ML models should also be updated periodically to ensure that they remain accurate. .
- **Disease Prediction:** Based on the quantity of contaminants in the air, the system should be able to assess the collected data and compute the likelihood of lung cancer and other associated diseases.

5.4 Non-Functional Requirements

- **Performance:** The system should be able to manage enormous volumes of data and process it in real time.
- **Maintainability:** The system should be simple to maintain and upgrade, with as little downtime as possible.
- **Security:** Because IoT devices are frequently attacked, the system must be secure and preserve data privacy.
- **Scalability:** The system should be scalable and capable of supporting an increasing number of devices and data points.

5.5 Data Requirements

- **Data volume:** Sufficient data volume is required to train the machine learning model effectively. Generally, more data leads to better accuracy and more robust models.
- **Data quality:** The data used in machine learning projects must be accurate and representative of the real-world scenarios that the model will encounter. Inaccurate or inconsistent data can lead to incorrect predictions and unreliable models.
- **Data diversity:** It is essential for ensuring that the model is not biased toward a certain sample of data. To generalize properly, the model needs to be exposed to a variety of data samples.

- **Data labeling:** Labeled data is required to train supervised learning models effectively. The labeling process involves adding tags or labels to the data that indicate the input/output relationship.
- **Data preprocessing:** Preprocessing involves cleaning and transforming the raw data to prepare it for machine learning algorithms. Data preprocessing can include tasks such as normalization, feature engineering, and data augmentation.
- **Data storage and management:** to handle massive amounts of data and make it available to machine learning algorithms.

CHAPTER VI
SYSTEM DESIGN

6.1 Current System

Current systems have many shortcomings that are listed as follows:

- They monitor air quality but do not predict the health effects of the monitored air on the human body.

  This is extremely harmful as the uninformed user will not be able to generate any useful inferences from it making the whole system redundant.
- They have modeled LSTM or Random Forest algorithms. This does not provide proper accuracy or efficiency.

  These models are not scalable for real time data or upcoming environmental factors.
- They focus on only one factor that affects lung cancer.

  They do not consider that there are multiple air pollutants that lead to lung cancer or lung cancer related diseases. Our research has shown that PM 2.5, PM 10 and CO are all pollutants of major interest as far as lung cancer is concerned.
- Furthermore, they have not used any cloud based technologies to test the project's feasibility.

  The system will use these platforms to make the system more scalable and reliable.

6.2 Design Considerations

6.2.1 Architecture:

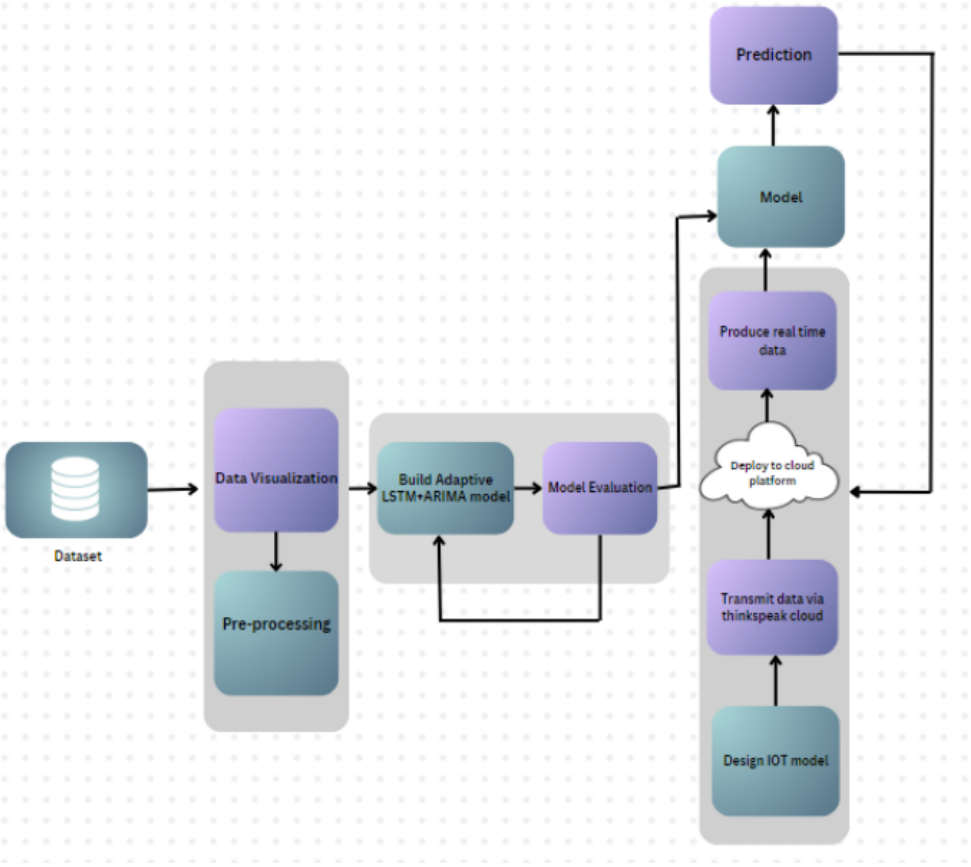

Fig 6.2.1: Architecture diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

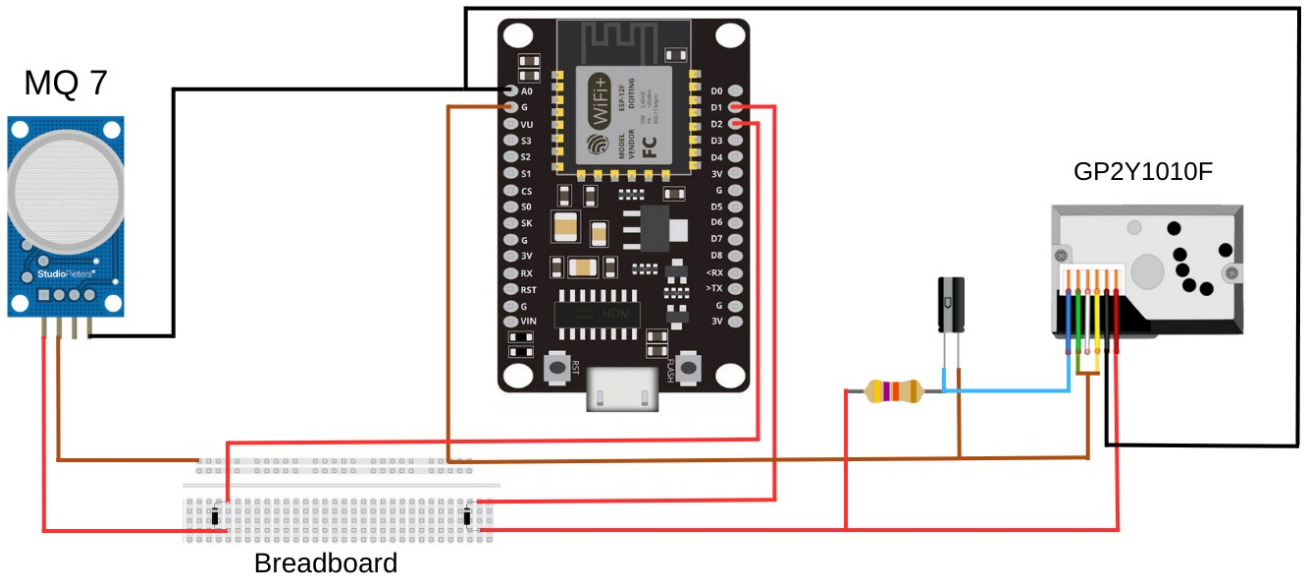

Fig 6.2.2: Circuit Architecture diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

The IoT station consists of the GP2Y1010AUF sensor for measuring the concentration of PM2.5 dust particles as well as the MQ-7 gas sensor for measuring the concentration of CO gas. The IoT station will consist of the ESP8266 WiFi module which will collect the data from the sensors and upload it to ThingSpeak cloud platform.

The system has 2 components wired up to the board.

- **MQ-7**: It receives power from the power line from ESP to the breadboard.
- **GP2Y1010AUF**: It receives power from the power line from ESP to the breadboard. Ground is set up in the same way.

6.2.2 Detailed flow

- **Pre-process the data**: Clean and preprocess the time series data, including handling missing values, transforming variables if necessary, and splitting the data into training and testing sets.
- **Train ARIMA model:** Fit an ARIMA model to the training data of the AQI dataset to capture the seasonal and trend components of the time series. Tune the ARIMA hyperparameters (e.g., order of differencing, autoregressive and moving average orders) using techniques like grid search or time series cross-validation.
- **Generate ARIMA forecasts**: Use the trained ARIMA model to generate short-term forecasts on the testing data.
- **Train Adaptive LSTM model**: Use the lung cancer dataset as input to train an Adaptive LSTM model on the training data, predicting the possibility of a person having cancer. This is done by converting the already existing values of "high", "moderate" ,"low" to 1, 2 and 3 using one-hot encoding.
- **Generate predictions**: Generate predictions using the trained Adaptive LSTM model to predict the possibility of a person having lung cancer and use the forecasts from the ARIMA model of future AQI. This will make sure the predictions of the LSTM model are not only accurate but also more localised to the area that is being considered in the dataset.
- **Evaluate and optimize**: Evaluate the hybrid ARIMA-Adaptive LSTM model's performance on the testing data using appropriate evaluation metrics, and fine-tune the model as needed by altering hyperparameters or model architecture.
- **Use sensors to get real time data:** Real time pollutant concentration data will be collected from an IoT station. The IoT station consists of the MQ-7 gas sensor for measuring the concentration of CO gas and the GP2Y1010AUF dust sensor to measure the concentration of PM2.5 (dust). The IoT station will be using the ESP8266 WiFi module to read values from the sensors and upload it to the ThingSpeak cloud platform.
- **Deploy the model and sensor station to cloud platforms:** The model will be uploaded to the cloud and use live reading to make predictions.

Table 6.2.3- Pros and Cons of Using the Hybrid Network:

PROS	CONS
Both ARIMA and adaptive LSTM work best with time series data.	Adaptive LSTM limits the accuracy in cases where there is limited data.
ARIMA is easy to interpret and reliable for small and stationary datasets.	Adaptive LSTM models can be highly complex, making them difficult to interpret.
While adaptive LSTM works best for large, real time, varying length, non-stationary data and learns complex patterns.	Selecting the appropriate ARIMA model parameters (such as the order of differencing and the number of autoregressive and moving average terms) can be challenging and require a good understanding of the underlying data.
Adaptive LSTM is capable of dynamically adjusting to changes.	
LSTMs can capture long-term dependencies and temporal patterns in the data, which can be useful for predicting lung cancer risk based on changing air quality conditions.	
ARIMA captures the short-term forecasts, and adaptive LSTM captures the long-term forecasts.	

6.3 DESIGN DESCRIPTION:

6.3.1 Use case diagram:

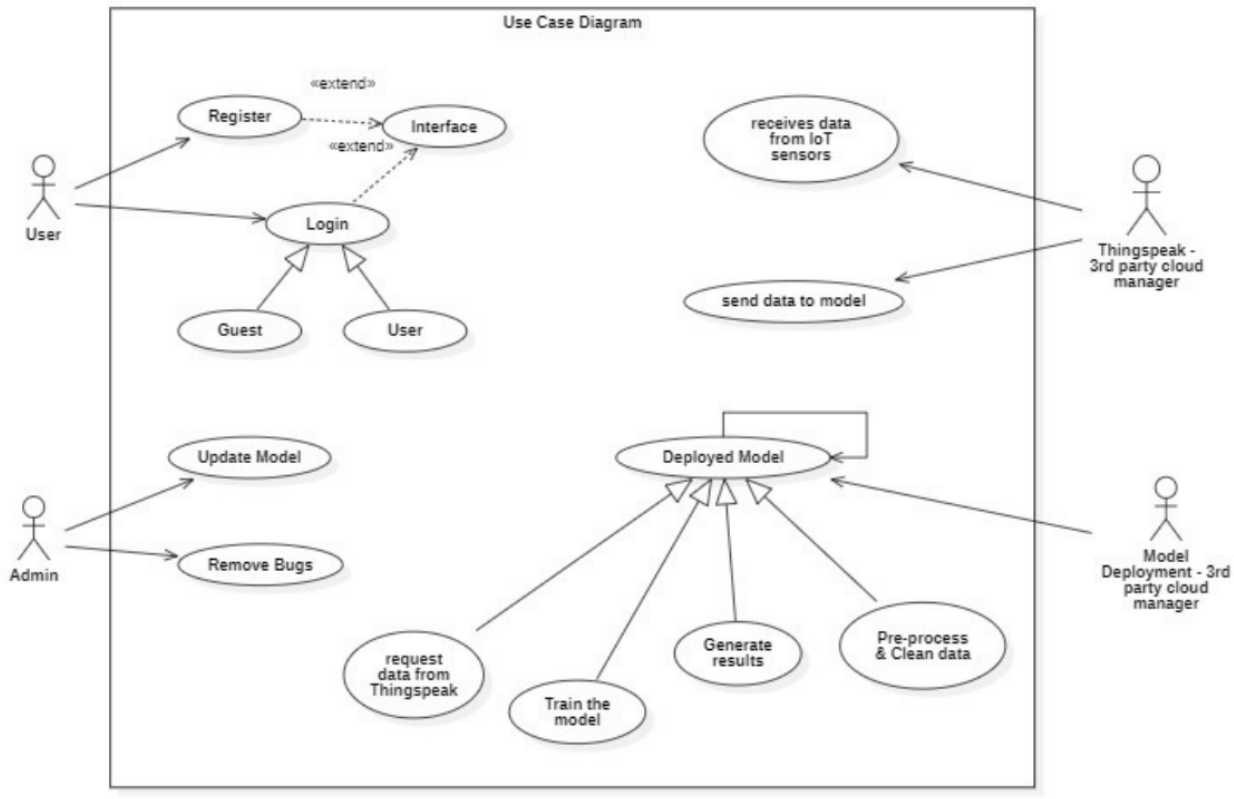

Fig 6.3.1: Use Case diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

Use cases that have been identified throughout the research:

- **User:** User will be an individual or entity that interacts with a system to see the predictions done on the air quality around them. Users will register and/or login in the system and will be the source of data from which the model will be further trained.
- **Admin:** Admin will be responsible for managing and maintaining a system or application. Admin will configure settings, manage user accounts, monitor system performance, and resolve technical issues. They will have the authority to access user details and patient details obtained from hospitals as well.
- **Cloud platforms:** Cloud platform refers to a set of cloud-based tools and services that will be used to deploy, manage, and scale applications and services. It will perform tasks such as deploying applications, scaling resources, managing data, and monitoring performance.

  The system will make use of two different cloud platforms: Thingspeak to receive data from the sensor stations and to send data to the cloud platform that hosts the model. Streamlit, Spaces or AWS to host the machine learning model.

6.3.2 Class Diagram:

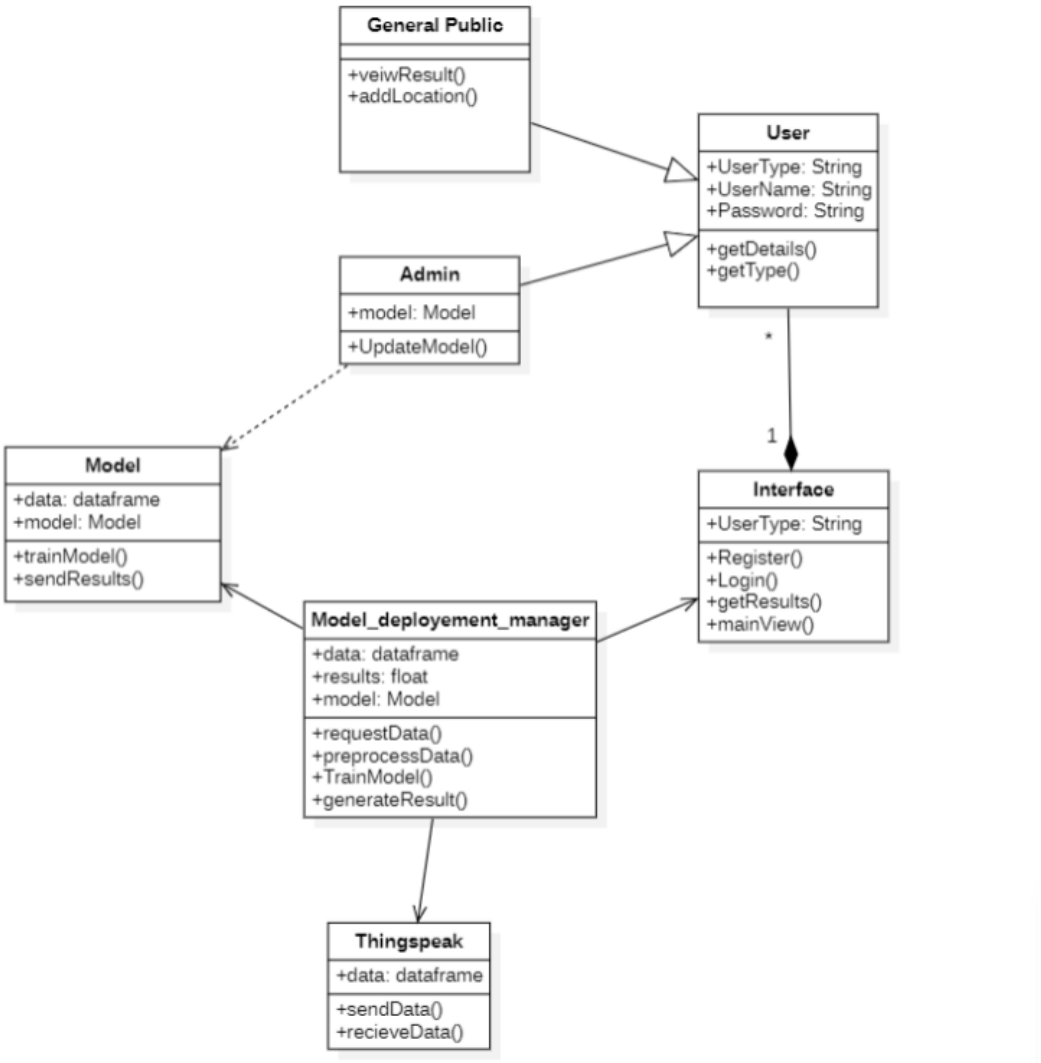

Fig 6.3.2: Class diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

6.3.3 ER DIAGRAM:

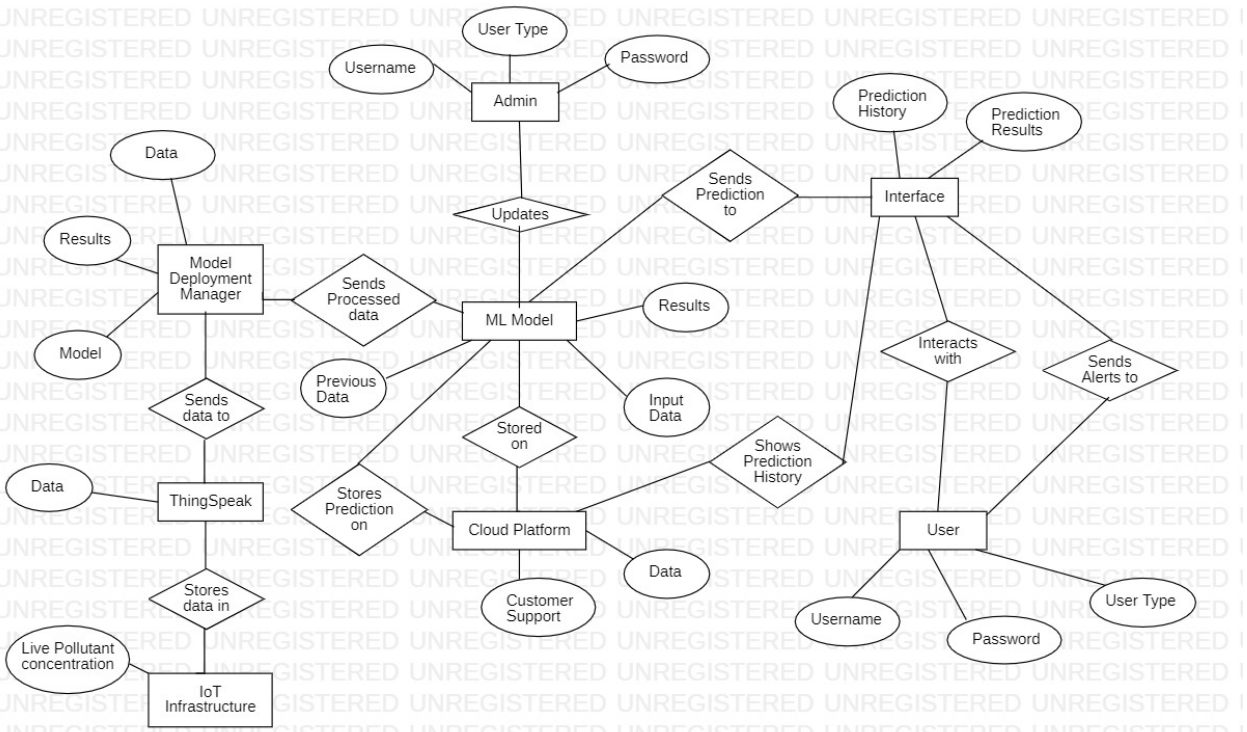

Fig 6.3.3: ER diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

6.3.4 Deployment Diagram:

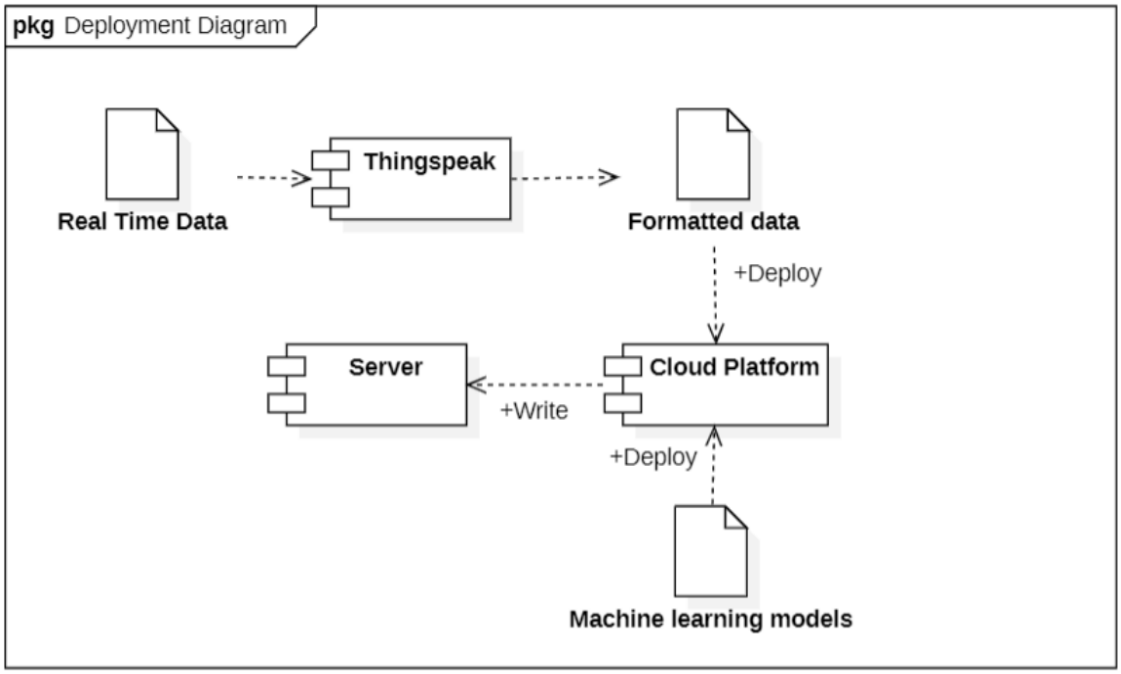

Fig 6.3.4: Deployment diagram for proposed system - Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

6.4 Design Details

6.4.1 Novelty

- The projects on this subject are based on outdated and static data.
- The system aims to use real time data to make predictions instead of static and outdated data.
- Most of the projects use just the LSTM model, The system will use an hybrid ARIMA + adaptive LSTM that is dynamically capable of adapting to changes.

6.4.2 Innovation

- Combining three domains- machine learning, cloud computing, and IoT
- Using a hybrid network of forecasting models- ARIMA+Adaptive LSTM
- By using a cloud platform for all the major functions, the entire system should be scalable, both horizontally and vertically.
- The system can be integrated into many devices due to the use of a cloud platform.

6.4.3 Reliability

- The project involves complex technical components, such as machine learning algorithms, cloud computing, and IoT devices.
- The system will be using tested and industry trusted cloud platforms for the deployment of the system.
- System modules will be designed that can help identify and replace any failed modules, thus improving overall reliability.
- Extensive testing will be conducted during the design and development phases to help identify potential problems and improve reliability.

6.4.4 Privacy and Security

- The dataset obtained does not contain the patient's personal details, thereby supporting privacy.
- A cloud platform will be used to store all the data as well as the model. Cloud platforms offer several security features to ensure that data stored on them is secure.
- Admin access will be given only to trusted individuals, thereby protecting the personal details of users and patients.

6.4.5 Performance

- The ML model should give an accuracy of 90%+ with a minimum error.
- The use of sensors that are known to give high amounts of accuracy in reasonable time periods will improve the performance of the whole system.
- A hybrid model will make sure to get the best of both worlds, providing us with the best features of both.
- The system will use the most relevant and informative features to avoid overfitting or underfitting the model.

6.4.6. Interoperability

- The data is collected and monitored on a cloud platform that can be accessed from any device.
- Adopting open standards for data formats, communication protocols, and APIs.
- Clearly defining the interfaces between systems, including the format of data exchanged and the methods of communication.

6.4.7 Maintainability

- The system should be easy to maintain and update, with minimal downtime.
- The system will implement a modular architecture that separates different components of the model, such as data processing, feature extraction, model training, and prediction. This allows for easier modification of each component without affecting the entire model.

6.5 Constraints, dependencies, and assumptions:

- **Interoperability requirement**s-

  Basic information can be displayed on an LCD connected to the stations. More data can be viewed on a computer with a basic internet connection. All error messages will be sent to the cloud platform. Some errors can be displayed on the LCD.

- **Interface/protocol requirements-**

  The accuracy of the data collected through IoT devices depends on the quality and reliability of the sensors used. There may also be biases in the data collected, which could affect the validity of the study results.

- **Discuss the performance related issues as releva**nt-

  The adaptive LSTM model might not have good accuracy if the amount of data fed is limited. The hybrid model might not give good accuracy if the data is irregular and contains anomalies that should be removed.

- **End-user environment-**

  Should provide real-time monitoring, continuously monitor the air quality using IoT devices, and update the database in real-time.

- **Availability of Resources**-

  Finding a skin cancer dataset and a global lung cancer dataset.

- **Hardware or software environment**-

  The amount of energy used by IoT technology is significant, and it needs to be running constantly**.**

CHAPTER VII
IMPLEMENTATION AND PSEUDOCODE

7.1 DATA PREPARATION:

```
[ ] df = pd.read_csv("/content/station_day.csv")

df.head(5)
```

	StationId	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	AP001	2017-11-24	71.36	115.75	1.75	20.65	12.40	12.19	0.10	10.76	109.26	0.17	5.92	0.10	NaN	NaN
1	AP001	2017-11-25	81.40	124.50	1.44	20.50	12.08	10.72	0.12	15.24	127.09	0.20	6.50	0.06	184.0	Moderate
2	AP001	2017-11-26	78.32	129.06	1.26	26.00	14.85	10.28	0.14	26.96	117.44	0.22	7.95	0.08	197.0	Moderate
3	AP001	2017-11-27	88.76	135.32	6.60	30.85	21.77	12.91	0.11	33.59	111.81	0.29	7.63	0.12	198.0	Moderate
4	AP001	2017-11-28	64.18	104.09	2.56	28.07	17.01	11.42	0.09	19.00	138.18	0.17	5.02	0.07	188.0	Moderate

Fig 7.1.1: Data preparation of the AQI dataset obtained from CPCB

Checking for null values:

```
df1= df.dropna()
print(df1.head())
```

```
        Date  PM2.5    PM10    NO    NO2    NOx    NH3    CO    SO2      O3  \
1 2017-11-25  81.40  124.50  1.44  20.50  12.08  10.72  0.12  15.24  127.09
2 2017-11-26  78.32  129.06  1.26  26.00  14.85  10.28  0.14  26.96  117.44
3 2017-11-27  88.76  135.32  6.60  30.85  21.77  12.91  0.11  33.59  111.81
4 2017-11-28  64.18  104.09  2.56  28.07  17.01  11.42  0.09  19.00  138.18
5 2017-11-29  72.47  114.84  5.23  23.20  16.59  12.25  0.16  10.55  109.74

   Benzene  Toluene  Xylene    AQI
1     0.20     6.50    0.06  184.0
2     0.22     7.95    0.08  197.0
3     0.29     7.63    0.12  198.0
4     0.17     5.02    0.07  188.0
5     0.21     4.71    0.08  173.0
```

Fig 7.1.2: Dropping null values from AQI dataset obtained from CPCB

```
lstm_df.head()
```

	Patient Id	Age	Gender	AQI	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Smoking	Passive Smoker	Clubbing of Finger Nails	Frequent Cold	Level
0	P1	33	1	2	5	4	3	2	3	2	1	2	Low
1	P10	17	1	3	5	3	4	2	2	4	2	1	Medium
2	P100	35	1	4	6	5	5	4	2	3	4	6	High
3	P1000	37	1	7	7	7	6	7	7	7	5	6	High
4	P101	46	1	6	7	7	7	6	8	7	2	4	High

Fig 7.1.3: Data preparation of the lung cancer dataset obtained from data.world

7.2 DATA VISUALIZATION:

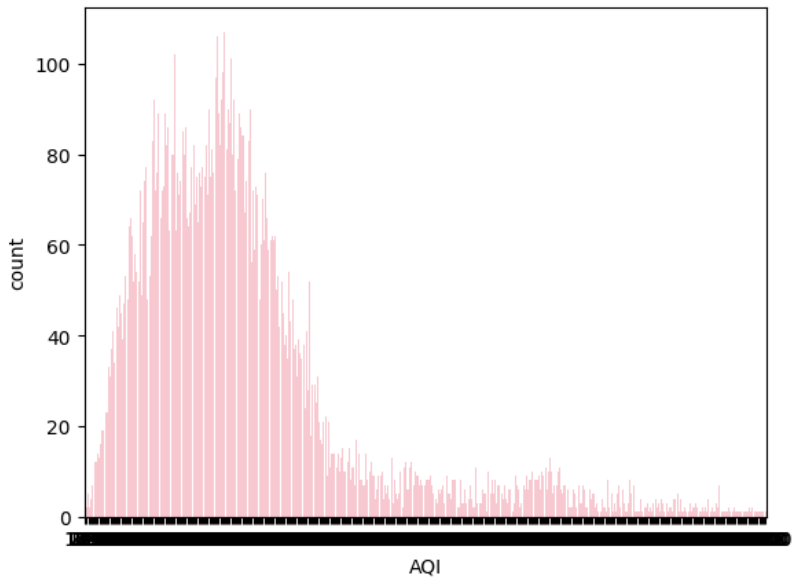

Fig 7.2.1: Count plot for AQI column in AQI dataset obtained from CPCB

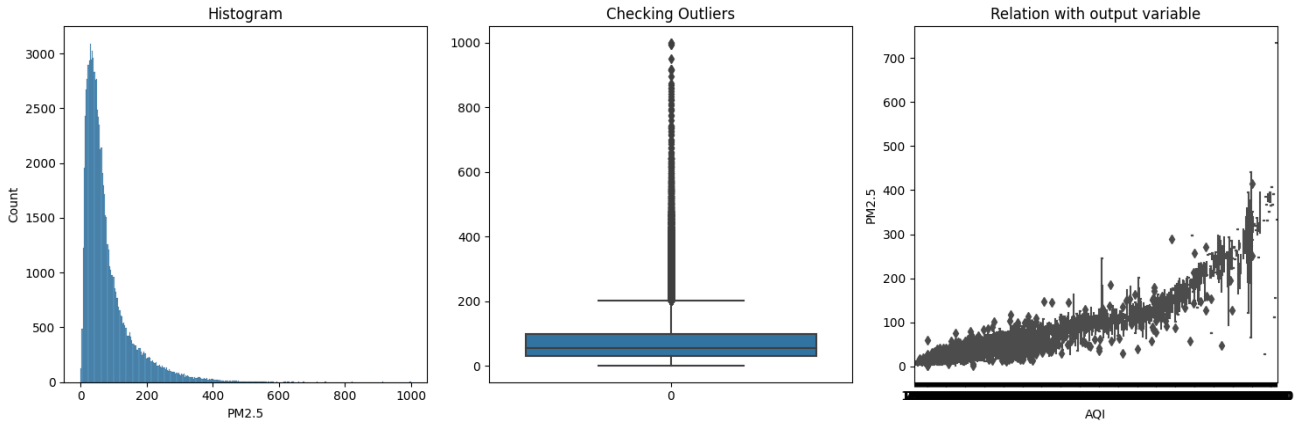

Fig 7.2.2: AQI vs. PM 2.5 and checking outliers in AQI dataset obtained from CPCB

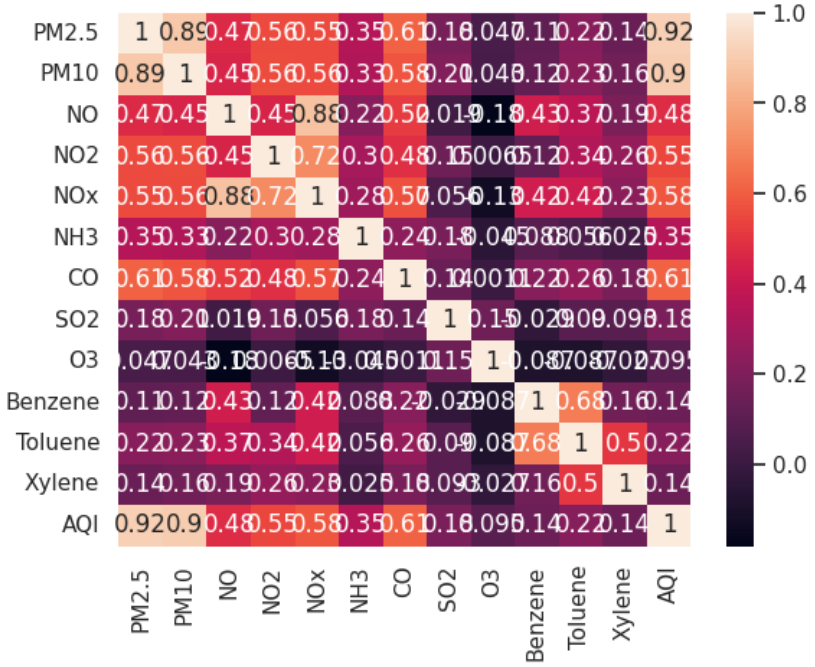

Fig. 7.2.3: Correlation matrix in AQI dataset obtained from CPCB

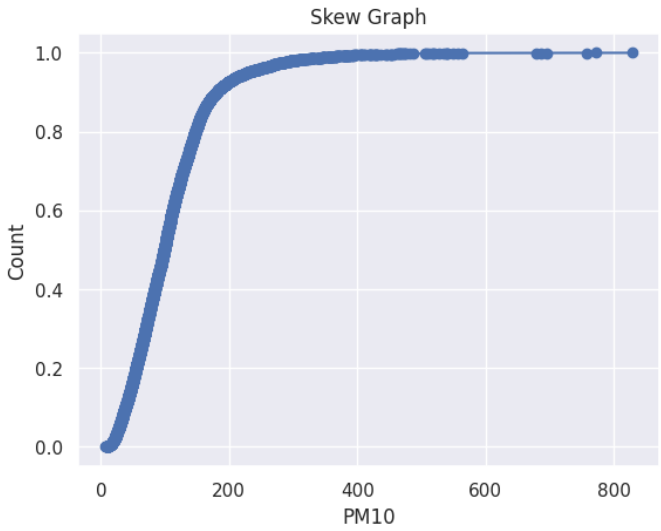

Fig 7.2.5: Skew plot for PM2.5 in AQI dataset obtained from CPCB

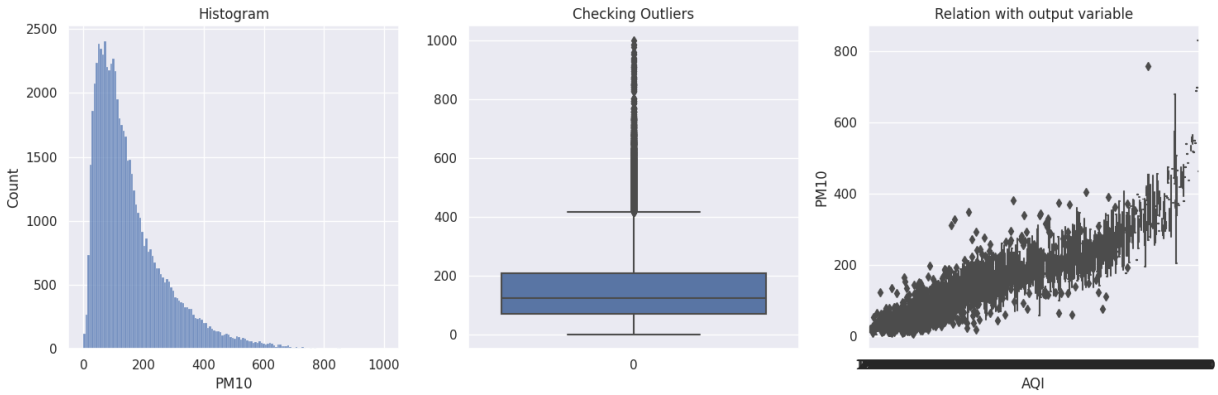

Fig 7.2.6: AQI vs. PM 10 and checking outliers in AQI dataset obtained from CPCB

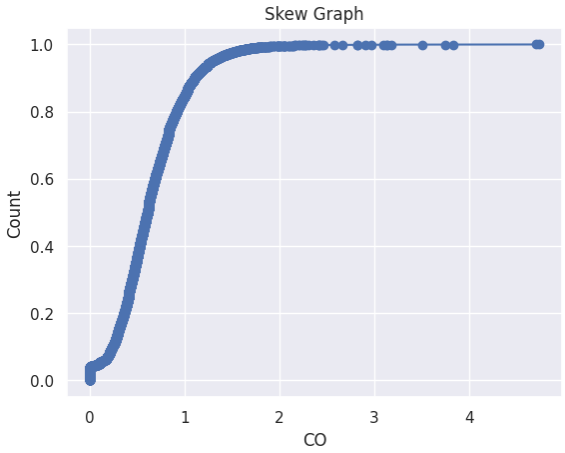

Fig 7.2.7: Skew plot for PM10 in AQI dataset obtained from CPCB

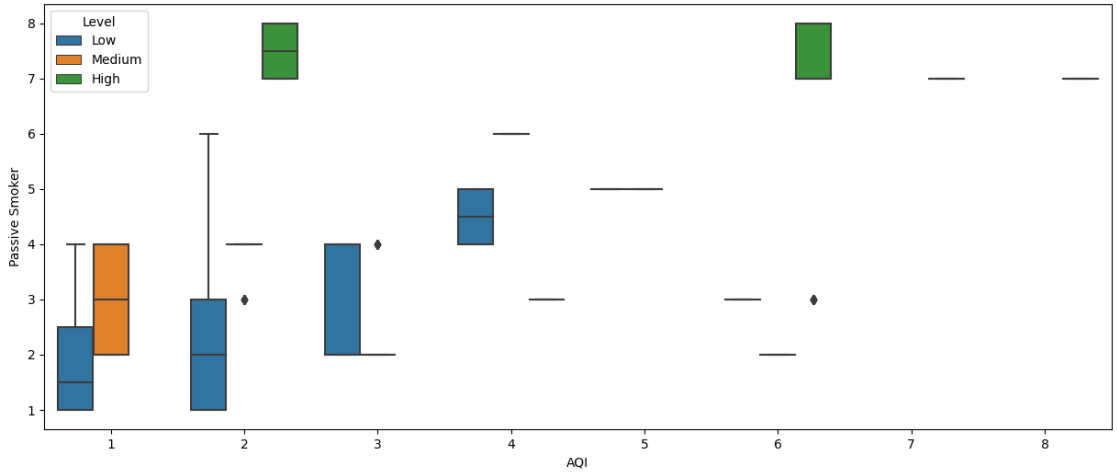

Fig 7.2.8: Relation of AQI and passive smokers of lung cancer dataset obtained from data.world

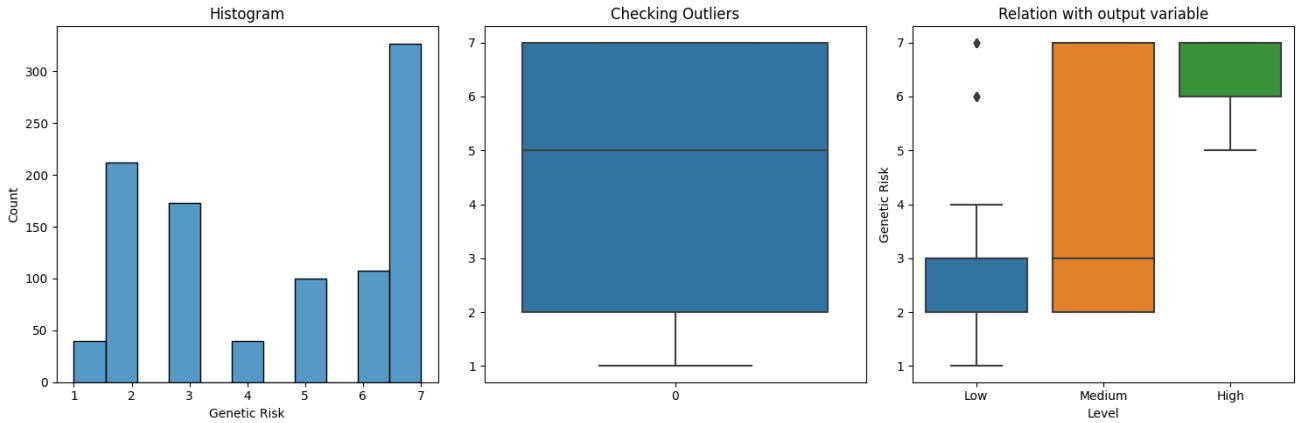

Fig 7.2.9: For checking outliers of lung cancer dataset obtained from data.world

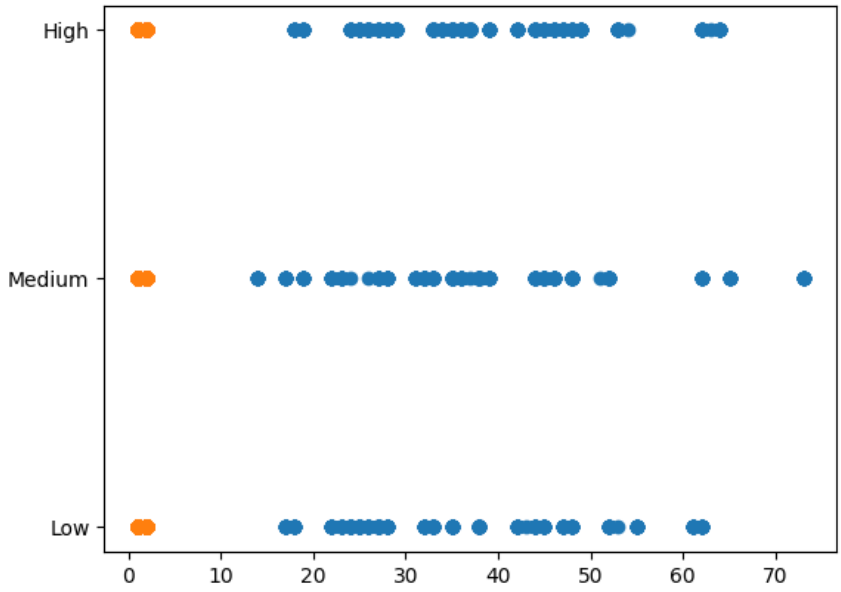

Fig 7.2.10: Scatter plot to check the risk level based on gender and age of lung cancer dataset obtained from data.world

7.3 ARIMA MODEL IMPLEMENTATION:

Autocorrelation plots:

ARIMA model works on stationary data, to check if the data is stationary, autocorrelation plots- ACF and PACF are unstructured.

ACF plots represent the correlation of time series with its lags. It measures the linear relationship between lagged values of the time series. PACF plots represent partial correlation of time series with its lags, after removing the effects of lower-order-lags between them.

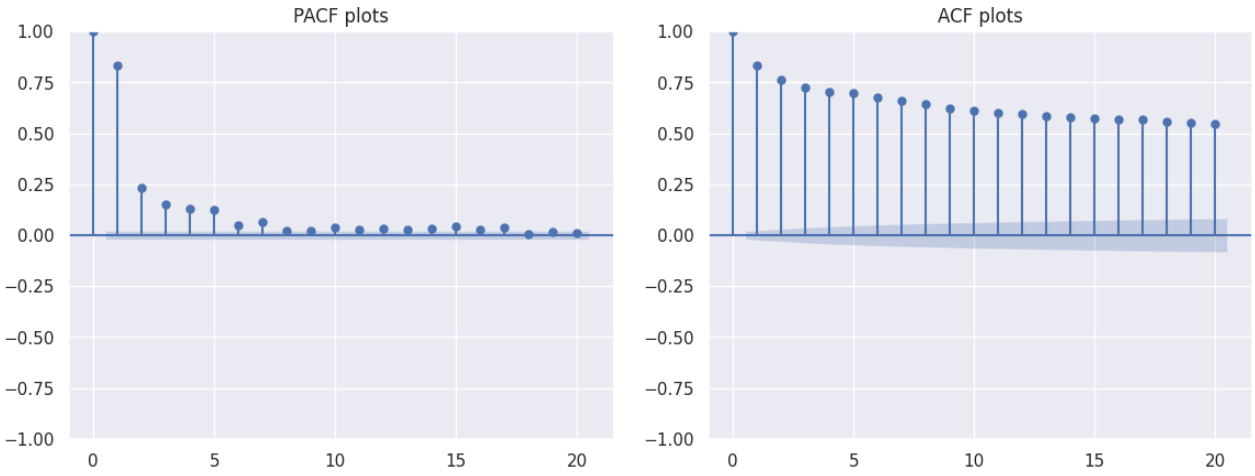

Fig 7.3.1 PACF & ACF plots for AQI dataset

The ACF plot shows the correlations with the lags are high and positive with very slow decay. While the PACF plot shows the partial autocorrelations have a single spike at lag 1.

These are both signs of a trended time series. So the time series is not stationary.

```
from statsmodels.tsa.stattools import adfuller
from numpy import log
result = adfuller(ts)
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
```

```
ADF Statistic: -1.224505
p-value: 0.662967
```

Fig 7.3.2 ADF test for AQI dataset

In our analysis, we employed the Augmented Dickey Fuller (ADF) test as another method to assess the stationarity of our data. The resulting p-value of 0.66 from this test indicates a significant value. In this context, a large p-value signifies that the test doesn't provide sufficient evidence to reject the null hypothesis. Consequently, the ADF test reinforces the observation that our time series data is non-stationary.

```
]  ts_train = ts[:50]
   ts_test = ts[50:]
```

Fig 7.3.3 Training, testing split for AQI dataset

The dataset for AQI was split into a 50:50 ratio. This was done to ensure an equal amount of data for evaluating model performance. This balance allowed the team to assess how well the model generalizes to unseen data, which is crucial for model validation. Another reason for this choice was to ensure that the testing set covers a similar time period as the training set. This is crucial for time series models as they rely on historical data to make accurate forecasts.

```
for i in pqd_combination:
    A_model = ARIMA(ts_train,order= i).fit()
    predict = A_model.predict(len(ts_train),len(ts)-1)
    e = np.sqrt(mean_squared_error(ts_test,predict))
    pqd.append(i)
    error.append(e)
```

Fig 7.3.4 p,q,d value selection

In ARIMA modeling, we focus on three vital parameters: p (autoregressive terms), d (non seasonal differences), and q (lagged forecast errors). A significant spike in PACF at 'p' without further spikes, along with a slowly decaying ACF, suggests an ARIMA(p, d, 0) model. Similarly, a substantial spike in ACF at 'q' without extending further, and a gradual PACF decay, implies an ARIMA(0, d, q) model. In our case, we choose 'p' as 2 and 'q' as 0, while 'd' is set to the minimum value to make the time series stationary. This approach ensures effective ARIMA parameter selection.

```
arima_predict = model_ts_fit.predict(start = len(ts_train),end = len(ts))

import matplotlib.pyplot as plt
residuals = model_ts_fit.resid[1:]
fig, ax = plt.subplots(1,2)
residuals.plot(title='Residuals', ax=ax[0])
residuals.plot(title='Density', kind='kde', ax=ax[1])
plt.show()
```

Fig 7.3.5 Fitting the model and generating residuals

Once the autoregressive and moving average components of the model have been taken into account, these residuals show the unexplained variation or noise in the time series data.

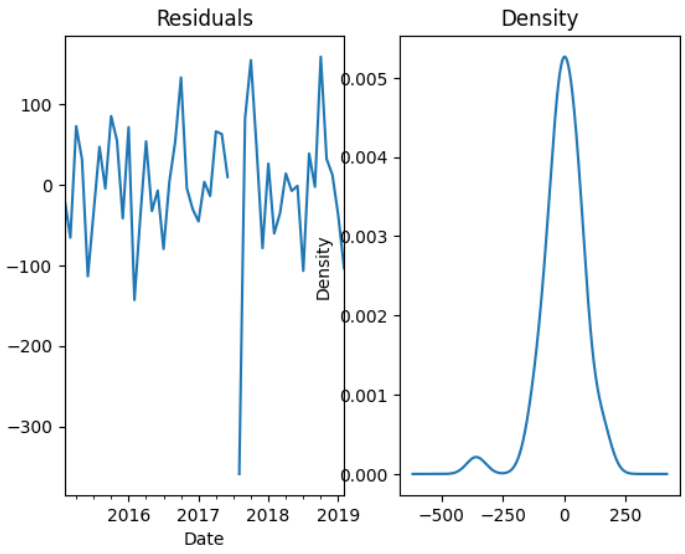

Fig 7.3.6 Residuals generated from ARIMA modelling of AQI dataset

The residuals look random in general, and their density looks normally distributed with a mean of around 0.

```
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error, mean_squared_error

forecast_test = model_ts_fit.forecast(len(ts_test))

rmse = np.sqrt(mean_squared_error(ts_test, forecast_test))
mae = mean_absolute_error(ts_test, forecast_test)
mape = mean_absolute_percentage_error(ts_test, forecast_test)
print(f'mae -: {mae}')
print(f'mape -: {mape}')
print(f'rmse -: {rmse}')
```

```
mae -: 69.24219899616345
mape -: 0.4782969407191707
rmse -: 79.14471630337096
```

Fig 7.3.7 Error calculation

The MAPE is a percentage-based metric that measures the average of the absolute percentage differences between predicted and actual values.

The error between the actual values and the predicted values is 0.47% which is very minimal.

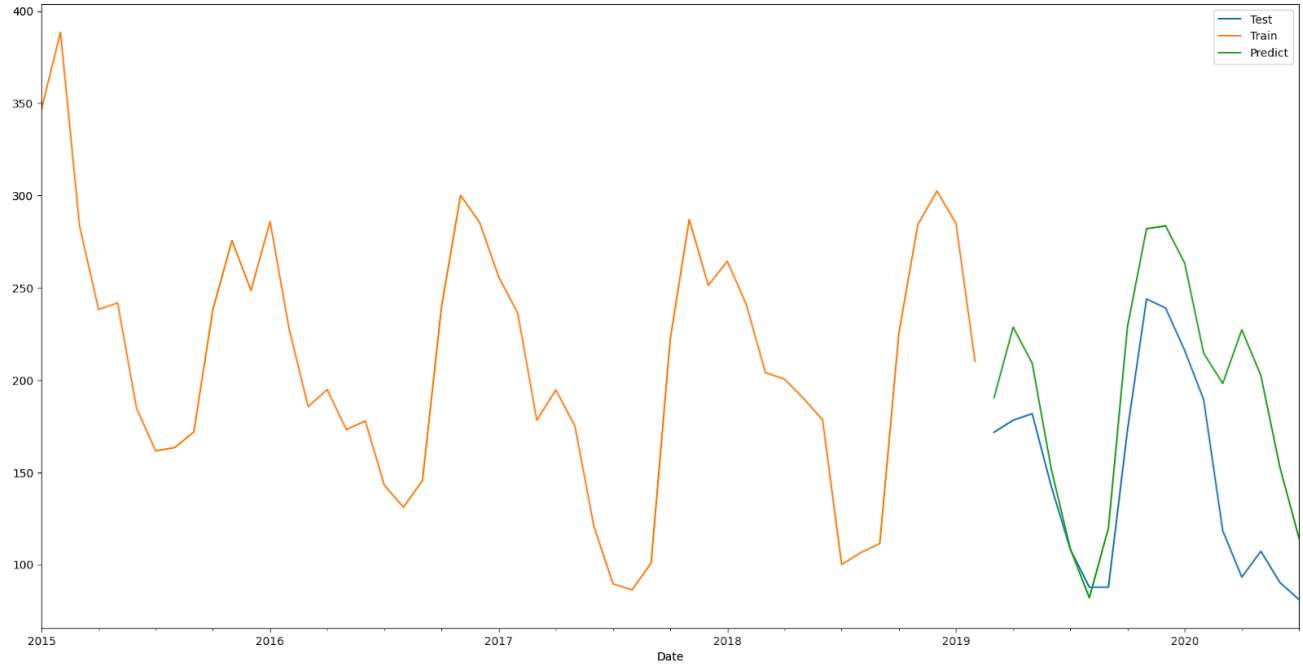

Fig 7.3.8 Plot of predicted values

7.4 LSTM MODEL IMPLEMENTATION:

```
import math
dataset  = lstm_df.values
training_data_len = math.ceil(len(dataset)*.8)
training_data_len
```

```
800
```

Fig 7.4.1 Training, testing split of data for lung cancer dataset

```
from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler(feature_range=(0,1))
scaled_data =  sc.fit_transform(lstm_df[['Age', 'AQI']])
scaled_data
```

```
array([[0.3220339 , 0.14285714],
       [0.05084746, 0.28571429],
       [0.3559322 , 0.42857143],
       ...,
       [0.18644068, 0.42857143],
       [0.06779661, 0.71428571],
       [0.55932203, 0.71428571]])
```

Fig 7.4.2 Scaling of lung cancer dataset to fit the model

A preprocessing tool called MinMaxScaler is used to scale numerical characteristics in the range of (0, 1). The fit_transform method applies the transformation after fitting the scaler to the data. The variable scaled_data contains the returned scaled data. The fit_transform method applies the transformation after fitting the scaler to the data (i.e., determining the minimum and maximum values). The variable scaled_data contains the returned scaled data.

```
model_lstm = Sequential()

model_lstm.add(InputLayer((12,1)))

model_lstm.add(LSTM(50))

model_lstm.add(Dense(34 ,'relu'))
# model_lstm.add(Dropout(0.25))

model_lstm.add(Dense(15 ,'relu'))

model_lstm.add(Dense(1 ,'relu' ))


model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(x_train.shape[1],1)))

model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

Fig 7.4.3 LSTM model initialisation

In our sequential neural network model, we've integrated an LSTM layer for sequence processing, followed by a Dense layer for the final output. Optimized with the efficient Adam optimizer, the model anticipates input data with 12 time steps and a single feature. The LSTM layer, with 50 units, is pivotal for handling sequential data and capturing temporal dependencies. Subsequent layers include a dense layer with 34 units and ReLU activation, another with 15 units and ReLU activation, and a final dense layer with 1 unit and ReLU activation.

```
x_test = np.array(x_test)
x_test = x_test.reshape((x_test.shape[0], x_test.shape[1], 10))
```

Fig 7.4.4 Reshaping test data

Reshaping ARIMA residuals is crucial before feeding them into an LSTM neural network to ensure adherence to the required input format. LSTM layers typically expect input data in a 3D tensor format with dimensions (batch_size, time_steps, features).

```
low_threshold = 1.10
high_threshold = 1.1

thatarray = []

categories = []

# Categorize the predictions
for prediction in predictions:
    per_cent = prediction[0] / (prediction[0]+prediction[1])
    thatarray.append(per_cent)
x = thatarray
x_norm = (x-np.min(x))/(np.max(x)-np.min(x))
print(x_norm)
```

```
[0.5034645  0.20249954 0.76791406 0.80944556 0.5788333  0.6824281
 0.6989642  0.8321171  0.54046166 0.8167146  0.865501   0.67782295
 0.69182265 0.67732704 0.97840536 0.8031117  0.42357558 0.68966883
 0.51739335 0.         0.21008034 0.4038655  0.5739447  0.7449166
 0.21674012 0.56863105 0.7432021  0.5743273  0.48653167 0.22172786
 0.44511357 0.5951993  0.5136384  0.48847294 0.24333669 0.3958596
 0.7993142  0.79816645 0.9215706  0.6504612  0.49354568 0.6367874
 0.5082964  0.75032943 0.6109985  0.48847294 0.63735425 0.67693025
 0.7329857  0.5170391  0.325847   0.6734162  0.66368157 0.6492993
 1.         0.16443966 0.6014765  0.3722812  0.7346861  0.7338784
 0.6649852  0.82619417 0.74619186 0.7693027  0.7168464  0.51062024
 0.9610758  0.62064815 0.80129796 0.9252405  0.98855084 0.6581412
 0.23891573 0.45710117 0.13916087 0.04201324 0.40239185 0.40229267
 0.88334066 0.6192453  0.41474786 0.8162045  0.8943647  0.43970072
 0.24089949 0.237102   0.6137333  0.6368158  0.6268828  0.72123903
 0.5430264  0.6726085  0.70086294 0.9803466  0.71584034 0.40511245
 0.7024783  0.66739404 0.54094344 0.9672963 ]
```

Fig 7.4.5 Predictions of LSTM model on the lung cancer dataset

The predictions generated are based on a percentage probability metric.

i.e. 0.472913 value generated by the model would mean that there is a 47% chance of the person contracting lung cancer in the current conditions they are based on their genetic history, AQI, etc.

```
loss, mae = model.evaluate(x_test, y_test)

7/7 [==============================] - 1s 15ms/step - loss: 0.1241 - mae: 0.3265
```

Fig 7.4.6 Error analysis

The mean absolute error is 0.32 which is quite low and proves the model is it suggests that the model's predictions are very close to the actual observed values in the dataset.

7.5 IoT STATION IMPLEMENTATION:

IoT station:

The IoT station consists of two sensors - MQ-7 gas sensor for measuring the concentration of CO gas in the atmosphere and GP2Y1010AUF sensor for measuring the concentration of PM 2.5 dust particles in the atmosphere. The ESP8266 WiFi module will take reading from the sensors and upload it to the ThingSpeak cloud platform**.**

- **MQ-7 gas sensor:** The MQ 7 gas sensor plays a critical role in detecting CO. We provide power to MQ 7 from digital pin 1. We take readings from it at analog 0 pin.

- **GP2Y1010AUF particulate matter sensor:** The GP2Y1010AU0F dust sensor is instrumental in particulate matter detection. We provide power to it from digital pin 2. We take its reading at analog pin 0.

This hardware configuration enables the system to gather data from diverse sources, covering gasses and particulate matter. The flexibility in the ESP board selection and dust sensor options allows for adaptability based on factors like cost and availability, making the project versatile and well-suited for various operational scenarios. The elaborate and delicate wiring ensures the efficient flow of data between the hardware components and the ESP board, thereby facilitating accurate and reliable air quality monitoring.

IoT Circuit Diagram:

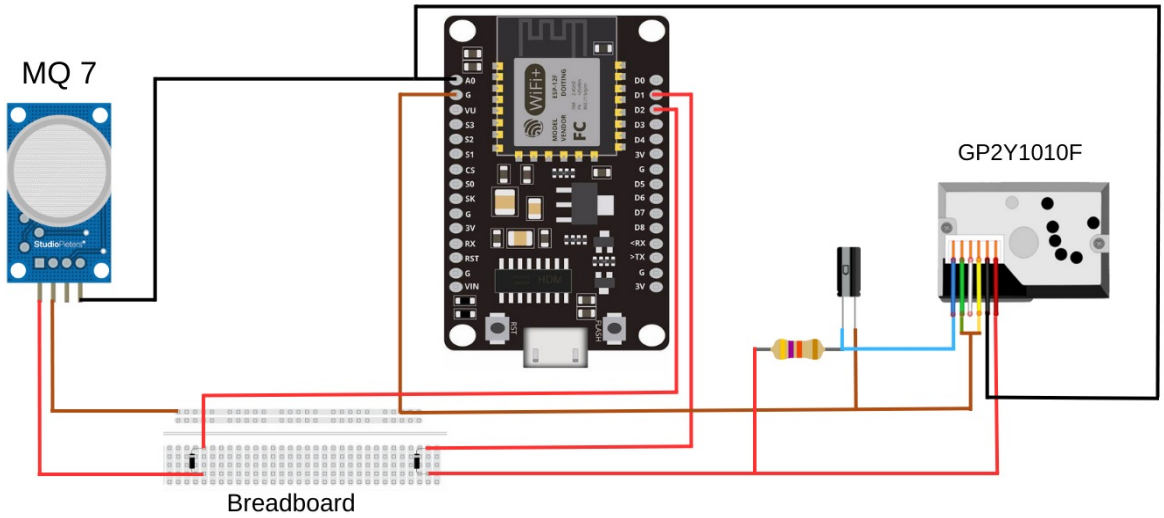

Fig 7.5.1 Circuit diagram

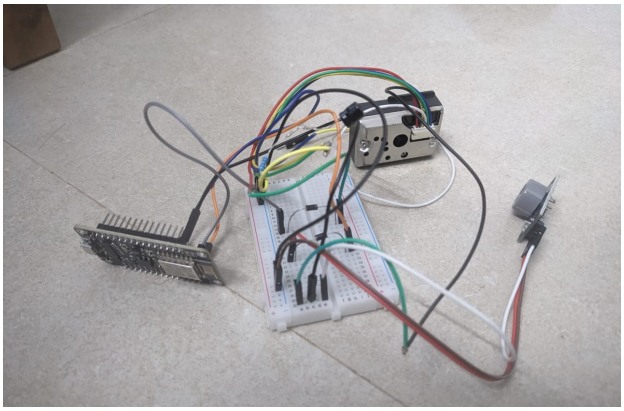

Fig 7.5.2 Real life diagram

Collection of data on ThingSpeak cloud:

The data pollutant readings collected by the sensors are analyzed and converted into meaningful values, and uploaded on the ThingSpeak cloud platform to be fed to the ML model..

final.ino

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include "MQ7.h"

MQ7 mq7(A0,5.0);

char ssid[] = "Phone";
char pass[] = "hereyougo2021";
WiFiClient  client;

const char * myWriteAPIKey = "ZL1K7BIINL8Z2EY0";
float dust = 0;
float co = 0;
float co_read = 0;
float calcVoltage = 0;
float dustDensity = 0;
int Pin_D1 = 4;
int Pin_D2 = 5;

```

```
20  void setup() {
21    Serial.begin(9600);
22    pinMode(Pin_D1, OUTPUT);
23    pinMode(Pin_D2, OUTPUT);
24    WiFi.mode(WIFI_STA);
25    ThingSpeak.begin(client);
26    if(WiFi.status() != WL_CONNECTED){
27      while(WiFi.status() != WL_CONNECTED){
28        WiFi.begin(ssid, pass);
29        Serial.print(".");
30        delay(5000);
31      }
32      Serial.print("Connected.");
33    }
34  }
35
```

```
35
36  void dustSensor() {
37    digitalWrite(Pin_D2,  LOW);
38    delay(1000);
39    digitalWrite(Pin_D1, HIGH);
40    delay(10000);
41  }
42
43  void coSensor() {
44    digitalWrite(Pin_D1,  LOW);
45    delay(1000);
46    digitalWrite(Pin_D2, HIGH);
47    delay(10000);
48  }
49
```

```
50  void loop() {
51
52    dustSensor();
53    dust = analogRead(0);
54    calcVoltage = dust * (3 / 1024.0);
55    dustDensity = 0.17 * calcVoltage - 0.1;
56    Serial.print("Dust Density = ");
57    Serial.println(dustDensity);
58    Serial.println("");
59    int x = ThingSpeak.writeField(2253626, 3, dust, myWriteAPIKey);
60    if(x == 200){
61      Serial.println("Channel update successful.");
62    }
63    else{
64      Serial.println("Problem updating channel. HTTP error code " + String(x));
65    }
66
```

```
  coSensor();
  co_read = analogRead(0);
  co = mq7.getPPM();
  Serial.print("CO = ");
  Serial.println(co);
  Serial.println("");
  int y = ThingSpeak.writeField(2303832, 1, co, myWriteAPIKey);
  if(y == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(y));
  }
}
```

Fig 7.5.3 Arduino IDE code for collecting readings from the sensors and connection to ThingSpeak cloud platform

7.6 CLOUD PLATFORM

Excitingly, our project leverages Streamlit as a pivotal element of our cloud-based system. With its intuitive interface, Streamlit seamlessly pulls from GitHub, ensuring our application is always up-to-date for easy maintenance and extension. This cloud-based approach simplifies deployment, making our system readily accessible via a dedicated link without requiring local software installation.

Furthermore, Streamlit not only offers hosting but also provides an interactive interface, enhancing user engagement with our IoT module and machine learning model. This seamless integration ensures a comprehensive and user-friendly solution for continuous air quality monitoring and health hazard prediction.

Following are various screenshots of our working application hosted on Steamlit.

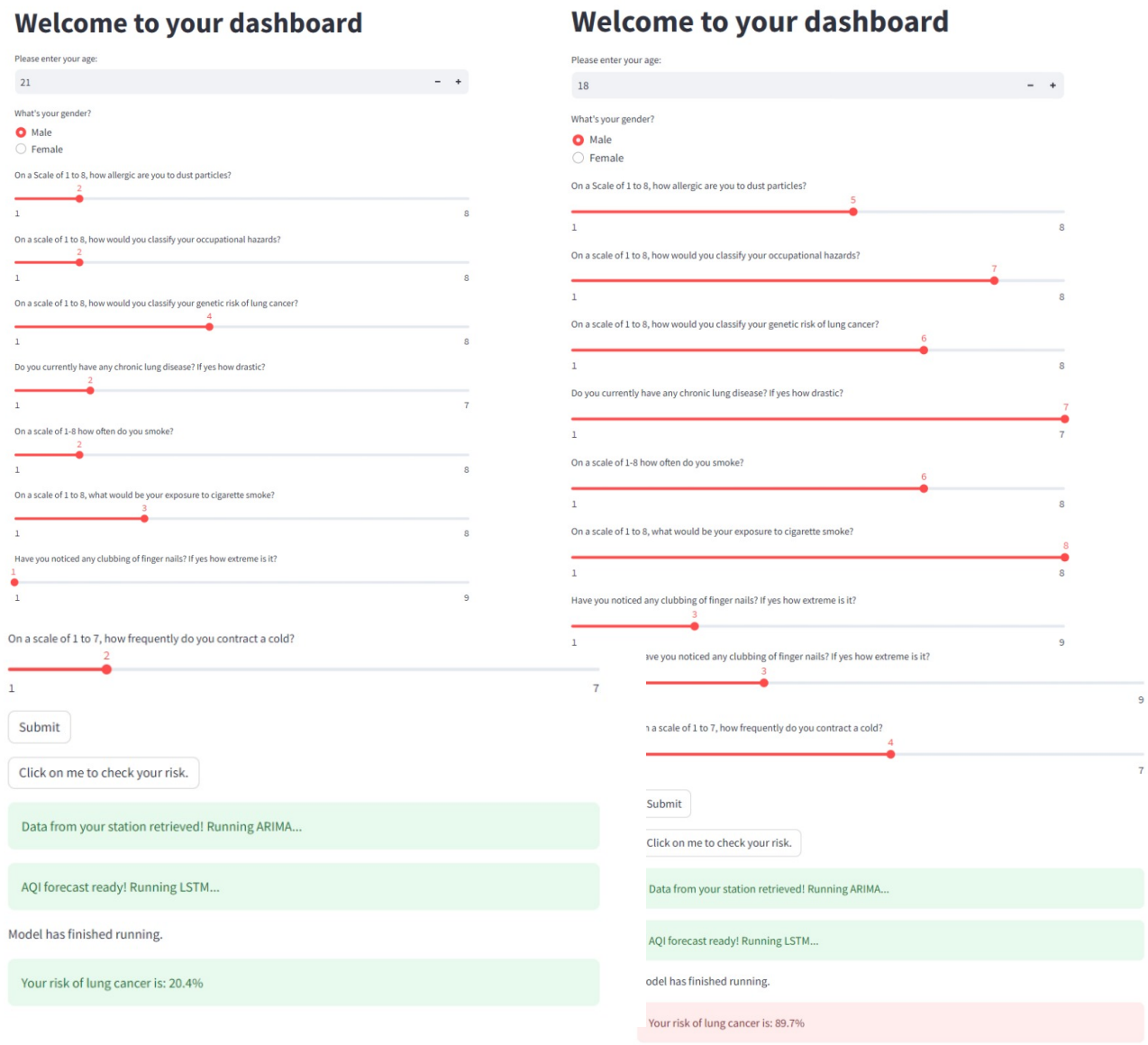

Fig. 7.6.1 Working application on Streamlit

CHAPTER VIII
RESULTS AND DISCUSSION

In this study, we harnessed the strengths of ARIMA and LSTM models to develop a hybrid prediction model for lung cancer incidence and have deployed this model on the cloud, alongside an IoT station to collect real time pollutant data. This unique approach capitalizes on ARIMA's ability to capture temporal patterns while leveraging LSTM's powerful sequential learning capabilities. It also combines the flexibility and scalability of cloud with the real life deployed IoT stations. To assess our hybrid model's performance, we compared it with two alternative models: the GRU model, a variation of LSTM focused on sequential learning, and the SARIMA model, renowned for its time series forecasting efficiency.

Following a comprehensive evaluation using various performance metrics such as Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), our hybrid ARIMA-LSTM model consistently outperformed both SARIMA and GRU models across all metrics. Our model yielded impressive results, with an MSE value of 686.16, RMSE of 79.144, MAPE of 0.47, MAE of 69.24, and a minimal loss of 0.124. In contrast, SARIMA obtained an MSE of 782.62, RMSE of 27.97, and MAPE of 311.37, while GRU achieved an accuracy of 1.13.

The IoT station deployed consists of the GP2Y1010AUF dust sensor to measure the concentration of PM 2.5 and the MQ-7 gas sensor to measure the concentration of CO gas. The ESP8266 has a shortcoming with only one analog pin. We are accepting analog inputs from two sensors. To

circumvent this, we are alternating power to the sensors. We achieve this by giving them power using digital pin 1 and 2 and changing their state in the code.

The data collected is uploaded to the ThingSpeak cloud platform and fed directly to the ARIMA model to generate forecasts. These forecasts are subsequently fed to the LSTM model to make predictions on the lung cancer incidence rate of the user.

The models are created in notebooks and they are then converted into regular python. This is then rewritten into a standard Streamlit application. Then we create a cloud website link using Streamlit community cloud. Then we link the github repository to the Streamlit cloud. Then Streamlit fetches the code and builds and hosts the application.

This research introduces a novel ARIMA-LSTM hybrid model for lung cancer risk prediction, demonstrating its superiority over traditional SARIMA and GRU models. Our comparative study underscores the enhanced prediction accuracy and potential to enhance early risk assessment and intervention strategies in the realm of lung cancer prevention. This innovative fusion of LSTM and ARIMA frameworks not only advances risk prediction techniques but also holds promise for applications in public health and personalized healthcare. The inclusion of an IoT station to collect live pollutant data broadens the horizon and allows us to extend the applications of our study into smart home systems as well. Governments can also use our system to identify pollution hotspots and create policies and regulations to reduce the harmful emissions and issue public health warnings in high risk areas.

CHAPTER IX
CONCLUSION AND FUTURE WORK

Accurate air quality monitoring and forecasting carry substantial theoretical and practical significance for the general populace. By alerting individuals to potential health hazards in their immediate surroundings, this research endeavors to offer valuable insights that could assist both government entities and the public. The study introduces a hybrid model, harmonizing ARIMA and Adaptive LSTM, harnessing air pollutant concentration data to predict air quality thresholds that may pose health risks. In essence, the research establishes that these time-series prediction models demonstrate proficiency, particularly when used in tandem as a hybrid system. Furthermore, the incorporation of a live Internet of Things (IoT) station for real-time air pollutant monitoring and a cloud hosted application for predictive capabilities augments the project's significance.

To enhance the project's efficacy, more comprehensive and granular data collection efforts could be undertaken. The integration of geographical data would unveil regional variations in air pollution effects. This versatile model can be further extended to investigate various health hazards linked to air pollution. Moreover, the IoT station could benefit from improvements, including the substitution of the ESP8266 module with a more potent counterpart featuring additional analog input pins. A deeper exploration of sensor technology and the incorporation of sensors capable of detecting a broader spectrum of gases would broaden the station's capabilities. The inclusion of a more dependable dust sensor would bolster data accuracy. While the project exhibits economic viability, exploration into cutting-edge technologies could further optimize its scope.

REFERENCES/BIBLIOGRAPHY

[1] An Application of IoT and Machine Learning to Air Pollution Monitoring in Smart Cities
By: Muhammad Taha Jilani, Husna Gul A.Wahab
[https://ieeexplore.ieee.org/document/8981707]

[2] How Is the Lung Cancer Incidence Rate Associated with Environmental Risks? Machine-Learning-Based Modeling and Benchmarking
By: Kung-Min Wang, Kun-Huang Chen, Shieh-Hsen Tseng
[https://www.mdpi.com/1660-4601/19/14/8445]

[3] Assessment of indoor air quality in academic buildings usng IOT and deep learnings
By: Mohammad Marzouk and Mohammad Atef
[https://www.mdpi.com/1667822]

[4] Household Ventilation May Reduce Effects of Indoor Air Pollutants for Prevention of Lung Cancer: A Case-Control Study in a Chinese Population.
By: Jin Z-Y, Wu M, Han R-Q, Zhang X-F, Wang X-S, et al.
[https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102685]

[5] Determination of Air Quality Life Index (AQLI) in Medinipur City of West Bengal(India) During 2019 To 2020 : A contextual Study
By: Samiran Rana.
[https://www.researchgate.net/publication/360622768_Determination_of_Air_Quality_Life_Index_Aqli_in_Medinipur_City_of_West_BengalIndia_During_2019_To_2020_A_contextual_Study]

[6] Air pollution and skin diseases: Adverse effects of airborne particulate matter on various skin diseases,
By: Kim Kyung Eun, Cho Daeho, Park Hyun Jeong
[https://pubmed.ncbi.nlm.nih.gov/27018067/]

[7] The spatial association between environmental pollution and long-term cancer mortality in Italy.

By: Roberto Cazzolla Gatti, Arianna Di Paola, Alfonso Monaco, Alena Velichevskaya, Nicola Amoroso, Roberto
[https://www.sciencedirect.com/science/article/pii/S0048969722055383#:~:text=We%20studied%20the%20links%20between%20cancer%20mortality%20and%20environmental%20pollution%20in%20Italy.&text=Tumor%20mortality%20exceeds%20the%20national%20average%20when%20environmental%20pollution%20is%20higher.&text=Air%20quality%20ranks%20first%20for,to%20the%20average%20cancer%20mortality.]

[8] The nexus between COVID-19 deaths, air pollution and economic growth in New York state: Evidence from Deep Machine Learning

By: Cosimo Magazzino , Marco Mele , Samuel Asumadu Sarkodie
[https://www.sciencedirect.com/science/article/pii/S0301479721003030]

APPENDIX A
DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- ARIMA - AutoRegressive Integrated Moving Average.
- LSTM - Long Short-Term Memory
- IoT - Internet of Things
- ML - Machine Learning
- WHO - World Health Organization
- AQI - Air Quality Index
- PM - Particulate Matter
- PMS3003- Plantover Particulate Matter Sensor
- INAAQS - Indian National Ambient Air Quality Standards
- ANN- Artificial Neural Network
- LCD - Liquid Crystal Display
- CNN - Convolutional Neural Network
- LSTM - Long ShortTerm Memory
- IAQ - Indoor Air Quality
- SVM - Support Vector Machine
- IDE - Integrated Development Environment
- OS - Operating System
- AWS - Amazon Web Services
- ROI - Return On Investment

APPENDIX B
SUPPORTING DATA

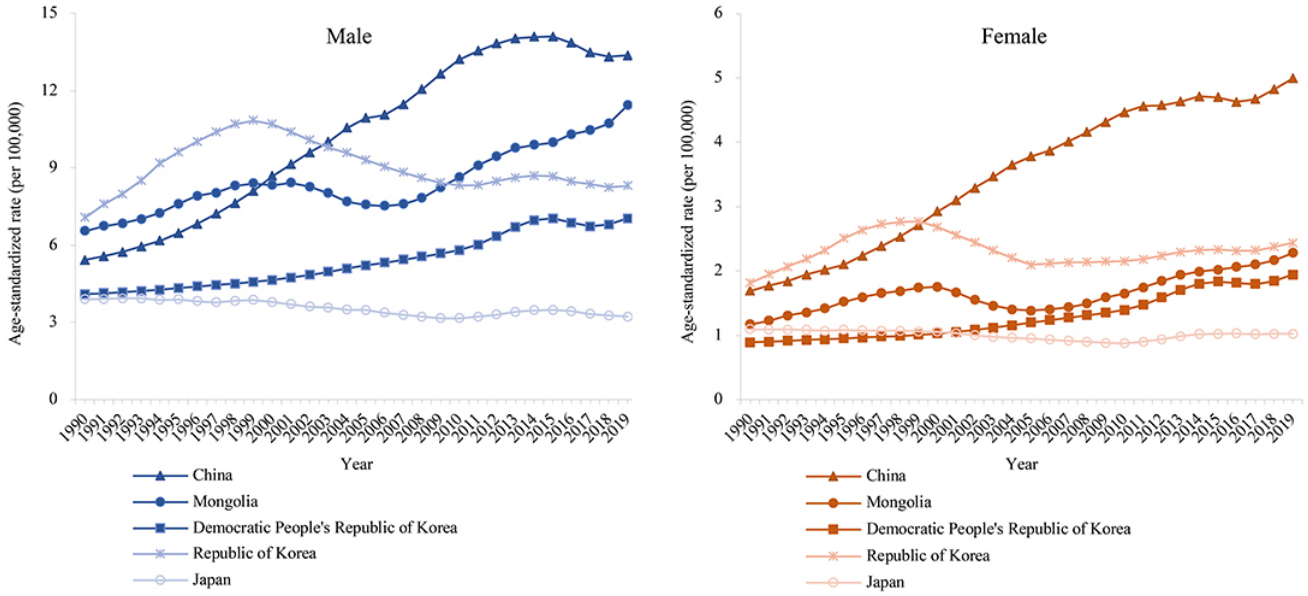

Fig 10.1.1: Lung Cancer death attributable to Long-Term Ambient Particulate Matter (PM2.5)

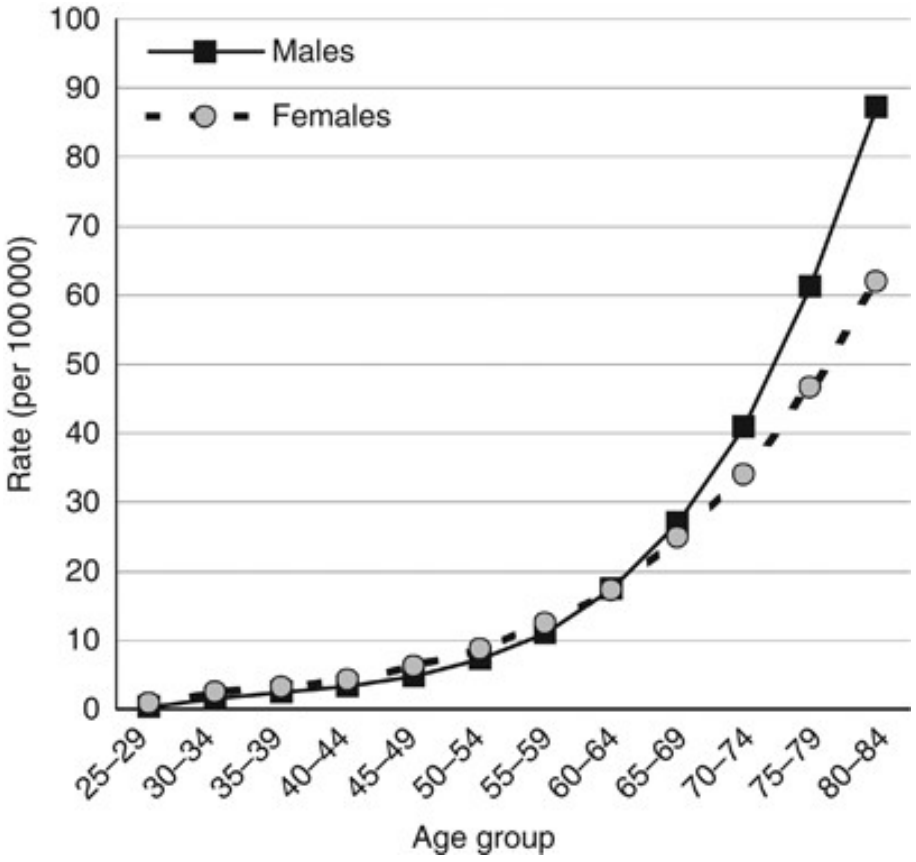

Fig 10.1.2 : Tobacco Attributable Cancer trends in Males and Females

Monitoring the concentration of air pollutants and its health hazards using Machine Learning models

Aditi Jain
in.aditijain@gmail.com
PES University

Aditya Shenoy
adityashenoy47@gmail.com
PES University

Ananya Adiga
ananyaadiga1@gmail.com
PES University

Anirudha Anekal
anirudhaanekal@gmail.com
PES University

Prof. Saritha
Dept of Computer Science, PES University

***Abstract*—With the world moving rapidly towards industrialization driven by economic growth and technological advancements, there is an alarming surge of air pollution leading to significant health concerns. In response, this paper introduces a research-driven approach for a continuous air quality monitoring system, meticulously designed to vigilantly track air quality in real-time in the proximity and proactively predict potential health hazards for the user. Central to the system's efficacy is a state-of-the-art hybrid Machine Learning model, seamlessly amalgamating the strengths of Adaptive Long Short-Term Memory (LSTM) and Auto-Regressive Integrated Moving Average (ARIMA) models, renowned for their ability in handling intricate time series data. This model is securely deployed on a cloud platform, ensuring not only accessibility but also scalability to meet current and future technological standards. The system primarily concentrates on the ongoing monitoring of air pollutants, such as PM2.5, PM10, and CO, ensuring that users have access to immediate and up-to-date insights into the air quality in their surroundings. Beyond this, the system goes a step further by employing this data to assess users' potential risk of developing lung cancer. Through the use of Internet of Things (IoT) sensors, the system can ready to issue timely and potentially life-saving insights, providing users with valuable information for decision-making enabling their well-being. In a world, where the link between air quality and health is increasingly evident, our research-based initiative serves as a beacon for a healthier future, while also fostering environmental consciousness and public well-being.**

***Index Terms*—Lung Cancer, IOT, Adaptive LSTM, ARIMA, AQI, Air Pollutants**

I. Introduction

Industrialization, while driving economic growth and technological progress, has ushered in a pressing concern of surge in air pollution. Yet, a critical issue persists among general public about awareness regarding the perils posed by compromised air quality. Pollutants such as PM2.5, PM10, and CO are particularly menacing to human health, with dire consequences for the lungs. The ease with which fine particulate matter infiltrates and effect the lungs, underscores the urgency of the health hazards awareness and hence encourages proactive steps needed to mitigate.

In response to this challenge, the development and deployment of air quality monitoring systems become imperative. Such systems hold the potential to furnish individuals with real-time insights into the air quality within their immediate vicinity, offering a proactive line of defense against potential health risks. By raising awareness and delivering early warnings, individuals gain the knowledge and time to undertake preventative measures, thus safeguarding their well-being.

Hence, this paper introduces a pioneering proposition - a continuous air quality monitoring system, dedicated to track air quality in the user's vicinity and informing them on the probability of possible health hazards. This research-driven approach seeks to enhance the efficiency of existing systems, leveraging appropriate data.

The envisioned solution is engineered to offer continuous surveillance of PM2.5, PM10, and CO levels in the atmosphere, providing users real-time insights about the air they breathe. Importantly, this solution gauges the user's potential health risks and subsequently issues timely alerts. By virtue of this approach, the solution empowers individuals to make informed choices and take measures to protect their health, constituting a critical stride in enhancing well-being in the face of escalating air pollution.

II. Related Work

Samiran Rana [1], in their paper, the research approach and findings unveil a distinctive path in the realm of air quality monitoring. Diverging from machine learning models, their method aligns with established guidelines from WHO and INAAQS, supported by empirical evidence showcasing a marked decrease in life expectancy attributed to air pollutants. Their precision-driven approach zeroes in on specific pollutant measurements, offering a more granular assessment than the Air Quality Index. Moreover, their investigation into the direct consequences of Particulate Matter (PM) on human health enhances their ability to communicate the tangible risks of airborne particulate matter exposure. Despite these advantages, the system's exclusive reliance on pre-established methods and its singular focus on PM2.5 pose limitations, emphasizing the need for continuous refinement in air quality monitoring systems for a more comprehensive and effective public health safeguard.

Cosimo Magazzino, et al. [2] proposed a comprehensive approach investigates the intricate dynamics of COVID-19-related mortality, employing machine learning models like

Artificial Neural Networks (ANN), Deep Learning, and Decision Trees. The research explores the interplay between COVID-19-related mortality rates, economic growth trends, and air pollutant concentrations (PM10, PM2.5, and NO2) within New York State. This meticulous analysis aims to uncover their potential influence on COVID-19 outcomes, offering valuable insights into this critical public health issue. The approach's advantages are threefold: it focuses on individual pollutants, meticulously designs data processes, and aligns methods effectively for real-time, personalized insights. However, limitations arise due to the focus on COVID-related deaths and complexities in attributing these deaths directly to air pollutants. The study encourages cautious interpretation and a broader perspective for examining air pollution's impact on COVID-19 outcomes.

In Iftikhar ul Samee, et al. [3] paper, the researchers adopt Artificial Neural Network (ANN) models to predict air contaminant concentrations and utilize Pearson's coefficient to assess their correlation with weather conditions. Their primary aim is to establish a pollutant monitoring system in smart cities, uncovering potential pollutant-weather correlations. The ultimate goal is to predict pollutant levels effectively to mitigate health risks, such as lung cancer. Notably, the study thoroughly considers weather conditions and demonstrates impressive model accuracy, with low Root Mean Square Errors (RMSE) for SO2 and PM2.5 predictions. However, limitations include the study's restricted consideration of environmental conditions, model simplicity, and IoT infrastructure complexity, highlighting the need for further research and system refinement in the context of smart cities.

M. Marzouk, et al. [4], and their team aimed to establish a correlation between outdoor pollution and indoor air quality (IAQ) by implementing real-time IAQ monitoring through Internet of Things (IoT) sensors strategically placed to collect data transmitted to the cloud. They employed Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models for data analysis. An advantage of their approach is its use of real-time data, offering a current IAQ understanding, especially concerning the interplay between outdoor and indoor environments, extending to rural areas. Nevertheless, the study has limitations, including the absence of family history data for individuals with lung cancer predisposition and the lack of consideration for occupational carcinogen exposures. Seasonal bias and a focus on select pollutants without accounting for additional substances like volatile organic compounds, NH3, and O3 are also noteworthy limitations.

Jin Z-Y, et al. [5] delved into the intricate relationship between home ventilation and lung cancer, utilizing a standardized questionnaire to gather a wealth of epidemiological and home ventilation data. Employing unconstrained logistic regression, they compute adjusted odds ratios (ORadj) alongside their corresponding 95% confidence intervals (CI). A notable strength lies in their comprehensive data collection approach, which encompasses a broad spectrum of variables, including family history, demographics, socioeconomic status, tobacco smoking history, dietary habits, and physical activity. However, potential limitations surface, such as localized data introducing biases and the study's narrow focus on indoor air quality (IAQ) without delving into the influence of outdoor pollutants. Furthermore, the omission of occupational exposures calls for further exploration of pertinent factors in assessing lung cancer risk.

K.-M. Wang, et al. [6], the researchers' approach and results leveraged a variety of machine learning models, including Logistic Regression, Random Forest, SVM, and Gradient Boosting. Benchmarking was a pivotal part of their investigation, yielding insights into the most effective model for their study. The primary objective was to establish a prediction model that uncovers the relationship between air pollution and lung cancer incidence rates. A notable advantage of their work was the comprehensive benchmarking, enhancing the study's credibility. Their dataset incorporated environmental risk factors and lung cancer incidence rates from diverse countries, bolstering the research's generalizability. Nevertheless, some limitations should be recognized, including a limited consideration of weather variables and the IoT infrastructure's complexity.

Kim KE, et al. [7] delve into the intricate relationship between Particulate Matter (PM) and skin-related ailments while shedding light on the underlying immunological pathways. They underscore that elevated levels of PM have been unequivocally linked to the development of various skin disorders, primarily through the modulation of oxidative stress and inflammatory cytokines. This notable correlation opens avenues for potential treatments, emphasizing the utility of antioxidant and anti-inflammatory medications in mitigating skin conditions induced by PM exposure. One of the commendable aspects of their paper is its meticulous cataloging of the air pollutants known to be responsible for specific skin diseases. The comprehensive listing encompasses a range of ailments, including Atopic dermatitis, Acne, Psoriasis, skin cancer, skin aging, alopecia, and oxidative stress. By providing this comprehensive overview, the paper equips readers with valuable insights into the potential health impacts of exposure to various air pollutants. However, it's essential to acknowledge that their paper's merits are balanced by certain limitations, such as the relatively limited use of statistical data to substantiate its claims. Furthermore, it falls short in elucidating the interplay between these skin conditions and their influence on the likelihood of one ailment triggering another, marking an area that warrants further exploration and research.

Roberto Cazzolla Gatti, et al. [8], in their paper, an AI-based approach was applied to investigate intricate links between cancer mortality, socioeconomic factors, and environmental pollution in Italy, spanning regional and provincial levels. The methodology included RF regression, Boruta feature analysis, and K-means clustering, with SMR forecasting. Notably, the research offers a detailed examination of the impact of air pollution on various body parts, emphasizing transparency in data sources and algorithms. However, it reveals limitations in

linking specific pollution sources to cancer and focuses solely on Italy, urging expansion to encompass global air pollution diversity.

III. Methodology

This section encompasses an exploration of the problem statement, elucidation of data collection methodologies, examination of the employed models, in-depth architectural insights, comprehensive evaluation processes, and the unveiling of resultant findings.

A. Proposed Work:

While specific air pollutants are currently under monitoring, a notable surge in diseases associated with air pollution remains evident. This research undertakes the crucial task of mitigating the impact of less-publicized air pollutants on public health by establishing a monitoring station to track their concentration. Furthermore, it endeavors to develop a Machine Learning model capable of forecasting disease incidence rates based on the concentration of these pollutants.

The hardware components for this project are meticulously selected to ensure robust and accurate air quality monitoring. At the core of the system lies the ESP8266, a versatile Wi-Fi-enabled module renowned for its ability to measure and transmit real-time data efficiently. Complementing this, the MQ-7 sensor plays a pivotal role in monitoring Carbon Monoxide (CO) levels, offering critical insights into this hazardous pollutant. Additionally, the GP2Y1010AU0F dust sensor is instrumental in tracking the presence of particulate matter with varying aerodynamic diameters, contributing to a comprehensive assessment of air quality. To bring these components together and facilitate their seamless operation, a reliable breadboard is employed to ensure the structural integrity of the hardware configuration.

The integration of these hardware elements forms the foundation of an effective air quality monitoring system, poised to offer real-time and accurate data, thereby enabling users to make informed decisions regarding their health and well-being.

The software ecosystem supporting this project is thoughtfully assembled to empower robust data analysis and seamless machine learning integration. It comprises a suite of statistical analysis tools, including Numpy and Pandas, to facilitate data manipulation and insights extraction. Complementing these are visualization tools like Matplotlib and Seaborn, which enhance the presentation of findings, making complex data more comprehensible. Machine learning capabilities are harnessed through Keras and TensorFlow, enabling the development and deployment of predictive models.

To ensure scalability and accessibility, the project leverages cloud computing platforms such as AWS, Streamlit and Thingspeak, which play a vital role in hosting and managing data. The Streamlit framework further augments the user experience by offering an interactive and user-friendly interface. Lastly, the Arduino IDE is employed for the development of Internet of Things (IoT) code, consolidating the software infrastructure to deliver a comprehensive solution for air quality monitoring and health prediction.

B. Dataset Description

In this paper, two key datasets are employed. The lung cancer patient dataset [9] is sourced from data.world, a platform, meticulously designed as a data catalog, provides open access to AI-ready datasets. This dataset forms the foundation of our analysis of the connection between lung cancer and air quality.

Simultaneously, the Air Quality Data in India (2015 - 2020) [10] dataset, featuring hourly data from different stations and cities in India, is gathered from kaggle.com. This dataset is publicly available through the Central Pollution Control Board, an official government portal, and plays a vital role in our examination of air quality trends over the specified time frame. The dataset has been compiled from Central Pollution Board's website and saved on kaggle.com.

C. Architecture and workflow

The architectural diagram and the flow of the project is described in this section.

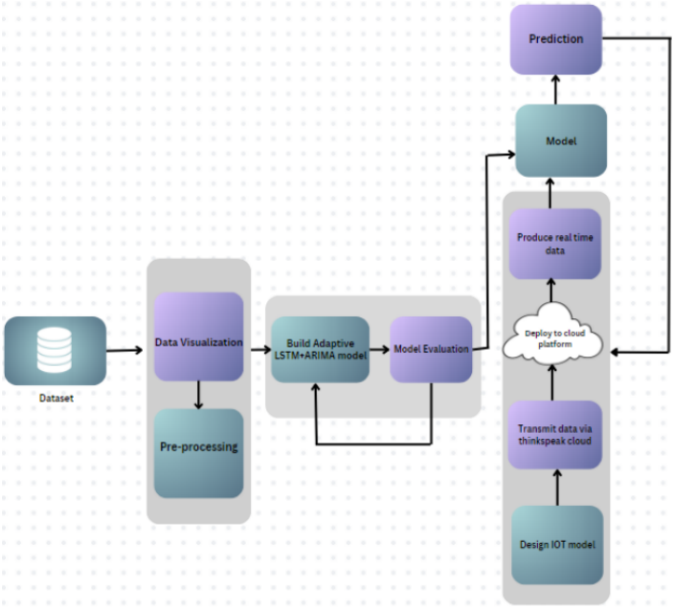

Fig. 1. Architectural diagram

Data Pre-processing:

In the initial phase, the the time series data from the sensor stations are cleaned and pre-processed. This involved handling missing values, addressing potential outliers, and transforming variables to ensure data quality.

The dataset was divided into distinct training and testing sets, with the training set designated for model training and the testing set reserved for evaluation purposes.

ARIMA Model Training:

The training data of the AQI dataset was fit to an ARIMA model. It was chosen for its capacity to capture seasonal and trend components in time series data.

Extensive hyper-parameter optimization took place, including determining the order of differencing, auto-regressive (AR) order, and moving average (MA) order. Optimization methods like grid search and time series cross-validation were employed to fine-tune the model.

ARIMA Forecasting:

With the trained ARIMA model in place, short-term forecasts of AQI were generated on the testing data. These forecasts provided valuable insights into expected air quality based on historical data.

Adaptive LSTM Model Training:

Using the lung cancer dataset as input to train an Adaptive LSTM model on the training data, the possibility of a person having cancer was predicted. This was done by converting the already existing values of "high", "moderate" and "low" to 1, 2 and 3 using binary encoding.

Generate Forecasts:

Then forecasts were generated using the trained Adaptive LSTM model to predict the possibility of a person having lung cancer and the predictions from the ARIMA model of future AQI.

Evaluation and Optimization:

Rigorous evaluation of the ARIMA-Adaptive LSTM model was undertaken using appropriate evaluation metrics on the testing data. This step ensured that the model met the desired standards for accuracy and reliability.

As necessary, the model underwent fine-tuning, which involved making adjustments to hyper-parameters and modifying the model architecture to optimize predictive capabilities.

Real-time Data Collection:

IOT sensor stations were constructed using ESP8266 modules. The system used MQ7 for CO detection, and GP2Y1010AU0F for dust detection. This allowed continuous and real-time collection of air quality data. More sensors can be connected for detection of other gases, however PM 2.5, PM 10 and CO was chosen after literature survey and economic considerations.

Deployment on Cloud Platforms:

To facilitate seamless data reception and transmission, the project deployed the machine learning model and storage and collection of the sensor stations' data on cloud platforms. Thingspeak was used to upload and temporarily store the sensor data.

The machine learning model can hosted on cloud platforms like Streamlit, Spaces, or AWS, ensuring scalability and accessibility while enabling real-time data processing and analysis. Streamlit was chosen to host the model, as it provides a basic interface for seamless and easy interaction with the model and other systems.

D. Research hypothesis

Within the scope of this study, as outlined in this paper, this research hypothesis revolves around implementing a comprehensive air quality monitoring system and constructing a machine learning model. This model aims to estimate the likelihood of an individual developing lung cancer and other associated health risks based on their exposure to harmful air pollutants. The evaluation encompasses a range of variables, including the individual's occupational environment, smoking history, lifestyle, and genetic predisposition.

To achieve this goal, the research entails a thorough analysis of pollutant concentration data acquired from wireless sensors, as well as the examination of the cloud-hosted trained model. Furthermore a basic interface to accept a user's details was also created. This multifaceted approach is central to the pursuit of enhancing public health awareness and contributing to the mitigation of air pollution-related health issues.

E. Machine Learning

The project is built on a combination network of Adaptive LSTM (Long Short Time Memory) and ARIMA (Auto-Regressive-Integrated Moving-Average).

Generally the pollutant concentration data will be a time-series data and the data used for training the model in this project is time series and stationary.

The data used for prediction is real time and dynamic, obtained after real time monitoring of various chosen pollutants using the IOT sensors. Hence, it is essential that the model be capable of handling large and dynamic data.

F. IOT

The application uses the IOT station to collect live pollution data and feeds it to the ML model for live risk analysis. In the IOT station MQ-7 Gas sensor has been used. It is primarily used for detecting carbon monoxide (CO) and methane (CH4) gases in the environment. GP2Y1010AU0F is a dust sensor designed to detect and measure the concentration of fine particulate matter (PM2.5) and larger particles in the air. The ESP8266 module used is a versatile and widely used Wi-Fi module that can be employed for various IOT (Internet of Things) and embedded electronics applications.

Wiring:

There were a few challenges when wiring up the IOT station. The biggest of which being the lack of multiple analog inputs on ESP8266.

ESP8266 provides only one analog input. The project takes analog readings from 2 sensors. To facilitate this, the sensors are programmatically switched off and switched on.

The power line to the sensors to MQ7 and GP2Y1010AUF are D1 and D2 pins respectively, from ESP8266. In the code saved on the module, D1 and D2 are set to output mode and set them to high and low as required. There is a diode between the power line and D1 and D2 pins. This is to ensure that only voltage above a threshold is applied to the sensors. The diodes

can be omitted, however it is not a useful move as removing them achieves nothing but a negligible economic gain.

As seen in Fig. 5, the station is shown without any enclosure. The station can be set inside one for cleanliness and protection from the elements. However if it is placed in an enclosure, it is recommended to solder the wires rather than using a breadboard for more secure wiring.

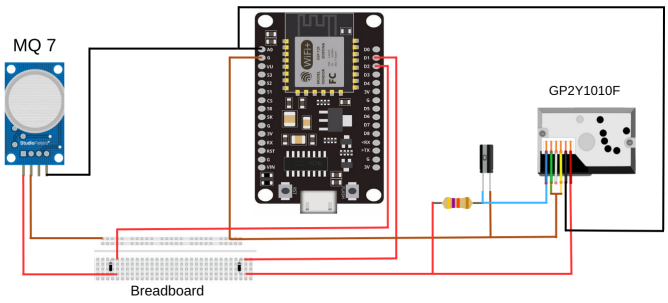

Fig. 2. *IOT circuit diagram with MQ7 and GP2Y1010AUF sensor*

The aforementioned IOT station has been built and has collected live data from multiple locations in the city to get a variety of readings and stored it on ThingSpeak. See below the station in real life.

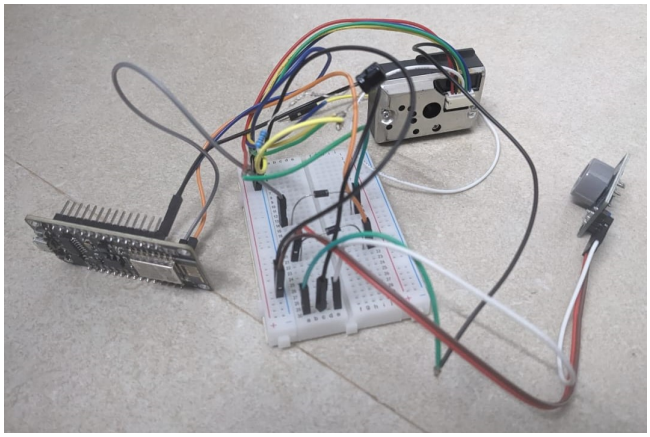

Fig. 3. *Live IOT station deployed to collect real-time data*

Collected data:

The concentration of the pollutants CO and PM2.5 was collected from various locations through the IOT station and uploaded on to ThingSpeak.

It was also used for temporary storage before moving the data to streamlit. Thingspeak provides API endpoints for read and write operations. The endpoint is hit by the code on streamlit. The ESP8266 code has a Thingspeak module that facilitates easy reading and writing.

ThingSpeak further allows us to visually represent the data in the form of graphs and helps us analyze the trends.

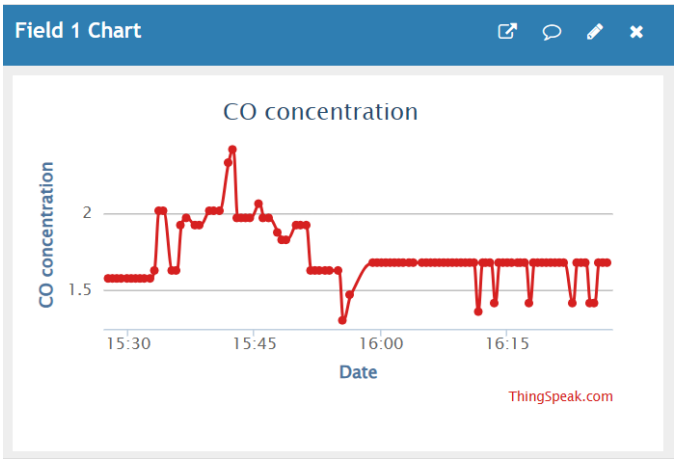

Fig. 4. Plot of concentration of CO in the atmosphere

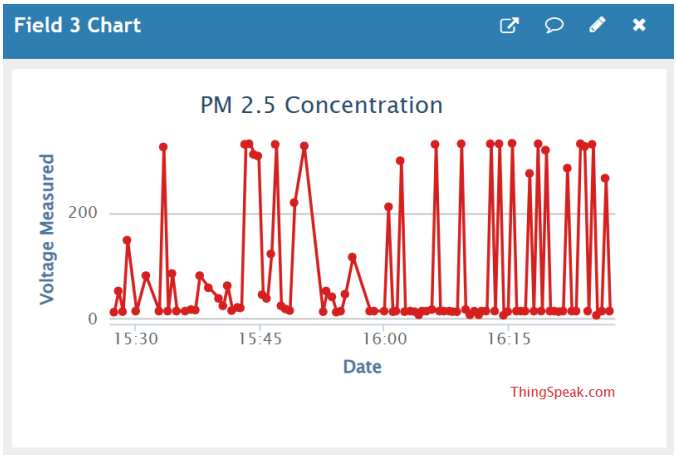

Fig. 5. Plot of concentration of PM2.5 in the atmosphere

G. Training and Validation:

Splitting data:

The data was split into 50% training set and 50% testing set for the ARIMA model. This was done to ensure an equal amount of data for evaluating model performance. This balance allowed the team to assess how well the model generalizes to unseen data, which is crucial for model validation. Another reason for this choice was to ensure that the testing set covers a similar time period as the training set. This is crucial for time series models as they rely on historical data to make accurate forecasts.

Autocorrelation plots:

ARIMA model works on stationary data, to check if the data is stationary, autocorrelation plots- ACF and PACF are unstructured.

ACF plots represent the correlation of time series with its lags. It measures the linear relationship between lagged values of the time series. PACF plot represent partial correlation of time series with its lags, after removing the effects of lower-order-lags between them.

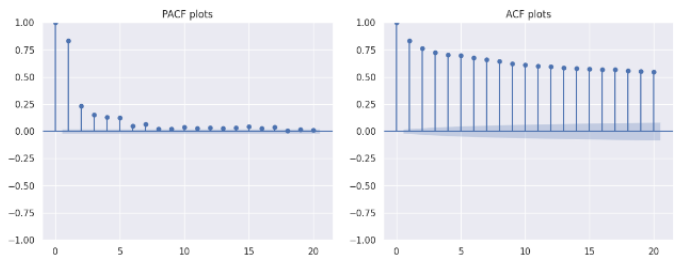

Fig. 6. ACF and PACF plots

The ACF plot shows the correlations with the lags are high and positive with very slow decay. While the PACF plot shows the partial autocorrelations have a single spike at lag 1.

These are both signs of a trended time series. So the time series is not stationary.

Fitting the model:

```
model_ts = ARIMA(ts_train, order=pqd[index])
model_ts_fit = model_ts.fit()
print(model_ts_fit.summary())
```

Fig. 7. ARIMA model fitting

'p' represents the number of autoregressive terms. Can be calculated as:

$$y_t = c + \varphi_1 + y_{t-1} + \varphi_2 + y_{t-2} + ... + \varphi_p + y_{t-p} + \varepsilon_t$$

c is a constant
$\varphi_1, ..., \varphi_p$ are parameters
ε_t is white noise.

'd' represents the number of nonseasonal differences needed for stationarity. It can be calculated by:

$$\nabla y_t = y_t - y_{t-1}$$

'q' represents the number of lagged forecast errors in the prediction equation. The formula is as follows:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q}$$

These values are set after analyzing the ACF and PACF plots- 'p' value is set to 2, since PACF shows more minor but significant lags at 2, 4, and 5.

In contrast, the ACF shows a more gradual decay. Values 2 or 3 or 4 can be used but to keep the model simple the value is taken as 2. The full equation of ARIMA(p,d,q) is:

$$\nabla y_t = c + \varphi_1 \nabla y_{t-1} + ... + \varphi_p \nabla y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + .. + \theta_q \varepsilon_{t-q}$$

'd' being the value of differencing, it is better to start with the lowest value, usually the value 1 makes it stationary.

If the PACF plot has a significant spike at lag p, but not beyond and the ACF plot decays more gradually. This may suggest an ARIMA(p, d, 0) model.

If the ACF plot has a significant spike at lag q, but not beyond and the PACF plot decays more gradually. This may suggest an ARIMA(0, d, q) model. Since the plots show the former, 0 is used as the 'q' value.

Residuals of ARIMA model:

The residuals of the ARIMA model are fed as the input to the adaptive LSTM model.

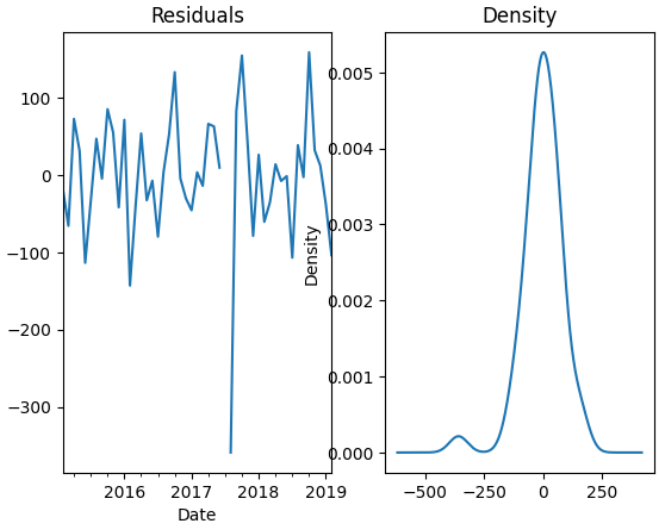

Fig. 8. ARIMA residuals

The residuals look random in general, and their density looks normally distributed with a mean of around 0.

These show that the residuals are close to white noise. We are ready to forecast with this model ARIMA(2, 1, 0).

Calculating error value:

The error factor MAPE is a metric that defines the accuracy of a forecasting method . It represents the average of the absolute percentage errors of each entry in a dataset to calculate how accurate the forecasted quantities were in comparison with the actual quantities.

$$M = \frac{1}{n} \sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}|$$

M = mean absolute percentage error
n = number of times the summation iteration happens
A_t = actual value
F_t = forecast value
The error value turned out to be 0.18.

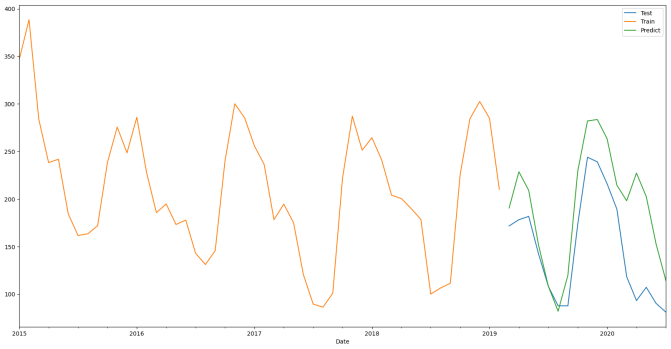

Fig. 9. ARIMA predictions

The error between the actual values and the predicted values is 0.47% which is very minimal.

H. Adaptive LSTM:

Defining Adaptive LSTM:

Importing sequential and dense layer libraries. Sequential neural networks is required for sequence processing. Dense layer is for the output. Both are required here because the input for prediction is real time.

```
model_lstm = Sequential()

model_lstm.add(InputLayer((12,1)))

model_lstm.add(LSTM(50))

model_lstm.add(Dense(34 ,'relu'))
# model_lstm.add(Dropout(0.25))

model_lstm.add(Dense(15 ,'relu'))

model_lstm.add(Dense(1 ,'relu' ))


model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(x_train.shape[1],1)))

model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

Fig. 10. LSTM layers initialisation

Splitting the data:

The input given is split into 80 percent training and 20 percent test data.

Training the model:

```
model.compile(optimizer = "adam", loss = "mean_squared_error", metrics=['mae'])

model.fit(x_train,y_train, batch_size=1, epochs=1)

740/740 [==============================] - 24s 27ms/step - loss: 0.0460 - mae: 0.1697
<keras.src.callbacks.History at 0x7d13b3fe7d00>
```

Fig. 11. Fitting the LSTM model on the lung cancer model

The 'compile' method defined, configures the model for training. 'Loss' specifies the loss function that will be used to measure the error during training. MSE is commonly used for regression problems, where the goal is to minimize the squared differences between predicted and actual values.

- optimizer='adam': Specifies the optimization algorithm that will be used during training.
- 'Adam' adapts the learning rate during training, which can lead to faster convergence.
- epochs=1: The entire dataset will be passed forward and backward through the neural network during training 1 time.
- batch size=1: This means that the model will update its weights after processing each individual sample.
- verbose=2: This means that it will display a progress bar for each epoch.

Calculate RMSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

After making the predictions and flattening them, the loss function Mean Squared Error is calculated.

Cloud platform:

In this project, Streamlit platform is used as a pivotal component of this cloud-based system. Streamlit offers an intuitive, user-friendly interface that seamlessly pulls from a GitHub repository to build this application. By doing so, it is ensured that the system is always up-to-date with the latest code, making it easy to maintain and extend as the project evolves. The cloud hosting provided by Streamlit, hosts this application on a custom chosen link, ensuring it's readily accessible to end users. Moreover, the project utilizes Streamlit to provide an interactive interface for users to input data, interact with the IoT module and machine learning model. This seamless inter connectivity between Streamlit, the IoT module, and the ML model streamlines the user experience and enhances the effectiveness of this project.

The application has 3 primary components. First the streamlit interface code. Second the ThingSpeak connection. Third the machine learning model.

The streamlit interface has been build using the standard streamlit library in python. All code is based on the official streamlit documentation provided to developers.

The IOT module is connected to this application using ThingSpeak. The IOT module sends values to ThingSpeak through a provided ThingSpeak C++ module. The application then sends a specific HTTP request to a chosen ThingSpeak endpoint to retrieve historic and latest reading from the IOT module.

The machine learning models are based off of various python notebooks created for this project. These notebooks have been converted to standard python and converted to work as per the application specifications.

Below are some screenshots of the working application:

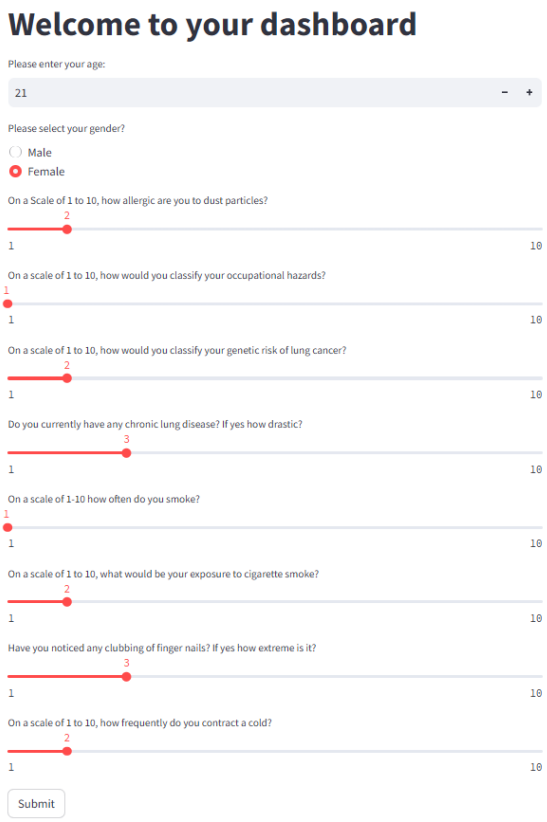

Fig. 12. Cloud interface to select

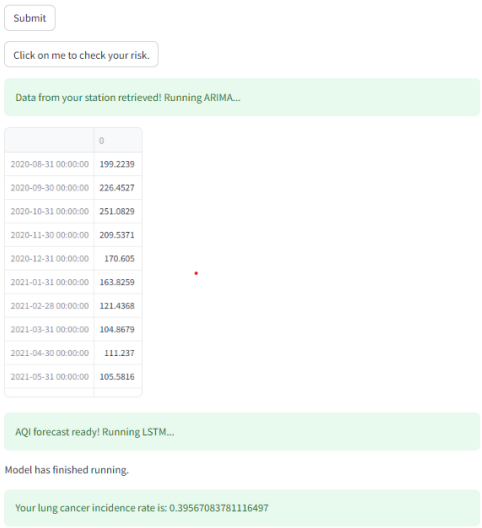

Fig. 13. Cloud interface

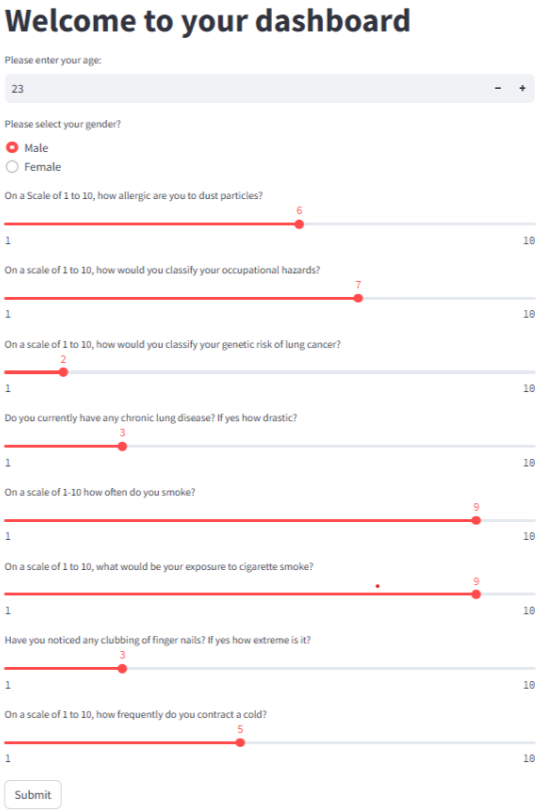

Fig. 14. Cloud interface

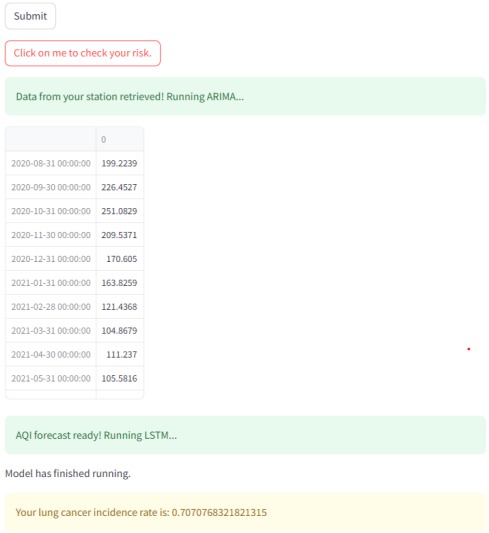

Fig. 15. Cloud interface

The user is asked 10 questions. They are first asked their age and sex. After that they are asked 8 questions where they are required to answer in a range between 1 and 10. They are asked about their dust allergy, occupational hazards, known genetic risk to the chosen disease, existing chronic lung diseases, smoking habits, passive smoke intensity, clubbing of finger nails and finally frequency of contracting colds.

After this data is submitted, data from ThingSpeak is retrieved. This pollutants data is sent to the ARIMA model to create forecasts. Then the forecasts and the user specific data is sent to the LSTM and finally a prediction is sent from LSTM model to the Streamlit website.

IV. Model benchmarking:

Following a comprehensive evaluation using various performance metrics such as Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), this paper's hybrid ARIMA-LSTM model consistently outperformed both SARIMA and GRU models across all metrics. The model in this paper yielded impressive results, with an MSE value of 686.16, RMSE of 79.144, MAPE of 0.47, MAE of 69.24, and a minimal loss of 0.124. In contrast, SARIMA obtained an MSE of 782.62, RMSE of 27.97, and MAPE of 311.37, while GRU achieved an accuracy of 1.13.

This paper introduces a novel ARIMA-LSTM hybrid model for lung cancer risk prediction, demonstrating its superiority over traditional SARIMA and GRU models. The comparative study in this paper underscores the enhanced prediction accuracy and potential to enhance early risk assessment and intervention strategies in the realm of lung cancer prevention. This innovative fusion of LSTM and ARIMA frameworks not only advances risk prediction techniques but also holds promise for applications in public health and personalized healthcare.

V. Conclusions and Future Work:

Accurate air quality monitoring and forecasting carry substantial theoretical and practical significance for the general populace. By alerting individuals to potential health hazards in their immediate surroundings, this research endeavors to offer valuable insights that could assist both government entities and the public. The proactive dissemination of information regarding air pollution may lead to a reduction in health-related issues and enhance emergency response capabilities, particularly during days of heightened pollution levels. The study introduces a hybrid model, harmonizing ARIMA and Adaptive LSTM, harnessing air pollutant concentration data to predict air quality thresholds that may pose health risks. Empirical findings demonstrate the efficacy of both models on time series data, with Adaptive LSTM exhibiting superior performance in the real-time and large data domains. In essence, the research establishes that these time-series prediction models demonstrate proficiency, particularly when used in tandem as a hybrid system. Furthermore, the incorporation of a live Internet of Things (IoT) station for real-time air pollutant monitoring and predictive capabilities augments the project's significance.

To enhance the project's efficacy, more comprehensive and granular data collection efforts could be undertaken. The integration of geographical data would unveil regional variations in air pollution effects. This versatile model can be further extended to investigate various health hazards linked to air pollution. Moreover, the IoT station could benefit from improvements, including the substitution of the ESP8266 module with a more potent counterpart featuring additional analog input pins. A deeper exploration of sensor technology and the incorporation of sensors capable of detecting a broader spectrum of gases would broaden the station's capabilities. The inclusion of a more dependable dust sensor would bolster data accuracy. While the project exhibits economic viability, exploration into cutting-edge technologies could further optimize its scope.

References

[1] Rana, Samiran. (2022). Determination of Air Quality Life Index (Aqli) in Medinipur City of West Bengal(India) During 2019 To 2020 : A contextual Study. Current World Environment. 17. 137-145. 10.12944/CWE.17.1.12.

[2] Cosimo Magazzino, Marco Mele, Samuel Asumadu Sarkodie, The nexus between COVID-19 deaths, air pollution and economic growth in New York state: Evidence from Deep Machine Learning, Journal of Environmental Management, Volume 286, 2021, 112241, ISSN 0301-4797, https://doi.org/10.1016/j.jenvman.2021.112241.

[3] I. u. Samee, M. T. Jilani and H. G. A. Wahab, "An Application of IoT and Machine Learning to Air Pollution Monitoring in Smart Cities," 2019 4th International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST), Karachi, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICEEST48626.2019.8981707.

[4] M. Marzouk and M. Atef, "Assessment of Indoor Air Quality in Academic Buildings Using IoT and Deep Learning," *Sustainability*, vol. 14, no. 12, p. 7015, Jun. 2022, doi: 10.3390/su14127015.

[5] Jin Z-Y, Wu M, Han R-Q, Zhang X-F, Wang X-S, Liu A-M, et al. (2014) Household Ventilation May Reduce Effects of Indoor Air Pollutants for Prevention of Lung Cancer: A Case-Control Study in a Chinese Population. PLoS ONE 9(7): e102685. https://doi.org/10.1371/journal.pone.0102685

[6] K.-M. Wang, K.-H. Chen, C. A. Hernanda, S.-H. Tseng, and K.-J. Wang, "How Is the Lung Cancer Incidence Rate Associated with Environmental Risks? Machine-Learning-Based Modeling and Benchmarking," *International Journal of Environmental Research and Public Health*, vol. 19, no. 14, p. 8445, Jul. 2022, doi: 10.3390/ijerph19148445.

[7] Kim KE, Cho D, Park HJ. Air pollution and skin diseases: Adverse effects of airborne particulate matter on various skin diseases. Life Sci. 2016 May 1;152:126-34. doi: 10.1016/j.lfs.2016.03.039. Epub 2016 Mar 25. PMID: 27018067.

[8] Roberto Cazzolla Gatti, Arianna Di Paola, Alfonso Monaco, Alena Velichevskaya, Nicola Amoroso, Roberto Bellotti, The spatial association between environmental pollution and long-term cancer mortality in Italy, Science of The Total Environment, Volume 855, 2023, 158439, ISSN 0048-9697, https://doi.org/10.1016/j.scitotenv.2022.158439.

[9] https://data.world/cancerdatahp/lung-cancer-data

[10] https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india