IoT Based Air Components Collection for Machine Learning Reinforcement

Prepared and Submitted by

Tanjima Islam
18101545
Fahad Rabbi
18101031
Rushana Ahmed
18101507
Md Muhtashemur Rahman
18101078
Mashrur Ahmed
18101409

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tanjima

Tanjima Islam
18101545

Rabbi

Fahad Rabbi
18101031

Rushana Ahmed

Rushana Ahmed
18101507

Muhtashemur

Md Muhtashemur Rahman
18101078

Mashrur Ahmed

Mashrur Ahmed
18101409

Approval

The thesis titled "IoT Based Air Components Collection for Machine Learning Reinforcement" submitted by

1. Tanjima Islam (18101545)
2. Fahad Rabbi (18101031)
3. Rushana Ahmed (18101507)
4. Md Muhtashemur Rahman (18101078)
5. Mashrur Ahmed (18101409)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 24, 2022.

Examining Committee:

Supervisor:
(Member)

Jannatun Noor Mukta
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

Hereby, we, the members, consciously assure that for the manuscript /insert title/ the following is fulfilled:

- This content is the writers' own unique work, which has never been published before.
- The work accurately and completely represents the authors' own research and analysis.
- The study appropriately acknowledges co-authors and co researchers for their significant contributions.
- The findings are suitably contextualized in light of previous and ongoing research.
- All sources are appropriately mentioned (correct citation). Text that is literally copied must be identified as such by using quote marks and providing suitable reference.
- All authors were personally and actively involved in significant effort leading up to the article and will accept public responsibility for its content.

Violations of the Ethical Statement standards may have serious repercussions. We agree with the preceding declarations and certify that this submission adheres to BRAC University's rules.

Abstract

Air pollution has been a noteworthy threat for a long time now in the 21st century. Human lives have never faced such an obscene amount of threat from the very air it needs to breathe to stay alive. As technology evolves more and more with every passing month, year, and decade, the emissions caused by the modern utilities are increasing as well. The measurement of air quality is done through an index called "AQI" which elaborates as the Air Quality index. The proposed work revolves around the collection of air component data through an IoT device and determining the AQI periodically and creating a proper dataset for the air quality index of the city of Dhaka. The IoT device is configurable to receive sensor data periodically. MQ-7, MQ-131, MQ-135 for air component detection, PMS5003 for particulate matter detection, DHT11 for humidity and temperature measurement and RTC DS3231 real-time clock module for timestamp has been used to make the device a complete frontrunner for a cheap data collection source. The data collection has been curated in such a way that pre-processing of datasets for certain machine learning and deep learning algorithm get much easier. All the sensors and modules are connected and worked in harmony by connecting them to a microcontroller (Arduino) and is stored and accessed remotely via an MPU (Raspberry Pi). The remote access is granted via cloud service (VNC Viewer). The acquired datasets are then ran through machine learning and deep learning layers (such as Random forest, Lasso Regression, Linear Regression, KNN, LSTM etc.) for the further prediction of the AQI.

Keywords:
Internet-of-Things(IoT); AQI; Dhaka; Air Quality Index; Time series Analysis; Multivariate; PM2.5; Machine Learning; Regression Analysis; LSTM; Deep Learning; Prediction; VNC Viewer; MQ Sensor; RTC; DHT11; Arduino; Raspberry Pi

Dedication

This thesis is dedicated to our institution's mentors, without whom we would not have been able to complete this thesis. They not only provided us with academic knowledge, but they also provided us with vital guidance when we needed it the most.

Acknowledgement

First and foremost, as a group we express our gratitude to Almighty Allah for allowing us to complete our Thesis on time and without any major obstacles.

Secondly, with that being said, we'd like to express our sincere appreciation to Jannatun Noor Mukta, our distinguished supervisor and instructor. Your enthusiasm and encouragement convinced us that working with you would be the right choice and it certainly has been so.

Thirdly, a special thanks to this group. This endeavor would not have been possible without this group.

Finally, we owe a debt to our parents. No amount of gratitude can make up for what they have done for us.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AQI	Air Quality Index
AWG	American Wire Gauge
$CASE$	Clean Air Sustainable Environment
GDP	Gross Domestic Product
$GPIO$	General-Purpose Input/Output
GPS	Global Positioning System
IC	Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of things
KNN	K-nearest Neighbors
LAN	Local Area Network
LDC	Least Developed Countries
$LPWA$	Low Power Wide Area
$LSTM$	Long Short-Term Memory
LTE	Long Term Evolution
MAE	Mean Squared Error
MCU	Microcontroller Unit
ML	Machine Learning
$MTMC - nLSTM$	Multiple Nested Long Short Term Memory Networks

PM	Particulate Matter
PWM	Pulse-Width Modulation
$RMSE$	Root-Mean-Square Deviation
$RMSLE$	Root Mean Squared Logaritmic Error
RNN	Recurrent Neural Network
RTC	Real-Time Clock
SEI	Sustainable Environmental Initiatives
SVM	Support vector machines
SVR	Support Vector Regression
UAV	Unmanned Aerial Vehicle
$US-EPA$	US Environmental Protection Agency
VOC	Volatile Organic Compounds

Chapter 1

Introduction

In 2015, Bangladesh attained the title of a lower middle income country from being one of the poorest and least developed nations in the world since its advent in 1971. Subsequently, the nation is on its way to advance towards being a developing country by 2026, moving up from the United Nations list of Least Developed Countries (LDC) [55]. In order to accomplish this, the nation has some challenges and problems that it has to address and tackle with finesse or else those issues will forever be a thorn in the path of development. One of these challenges is to overcome the severe air pollution that clutches the various populated areas of Bangladesh, especially the capital city of Dhaka. This severe air pollution and it being on the rise, only creates difficulty in the stride to become a developed country in the near future.

According to current WHO figures, 7 million deaths worldwide each year are due to pollutants in the environment [22]. The gravity of the pollution level in Bangladesh can be realized by statistics alone. As of 2021, Bangladesh was placed 1 in a global ranking of countries with most air pollution by IQAir, whose headquarters are based in Switzerland. This ranking was based on the AQI (Air Quality Index) of the countries, where Bangladesh scored the highest average AQI, which was 161. The $PM_{2.5}$ (Particulate Matter) concentration that was present in Bangladesh's air was 15.4 times more that the WHO annual air quality guideline value [59]. In Dhaka, tiny particles with a diameter range of less than 2.2 m account for 30–50% of the PM_{10} mass [12]. AQI is a metric that is used to measure the air quality of an area and the greater that of an area is, the more health concerns there are. In the main, $151 - 200$ AQI, where Bangladesh's AQI falls, means that the air is unhealthy for the general population and they may experience health effects and sensitive population groups like children and elders may experience severe health effects due to the polluted air [54].

Dhaka is a rapidly growing city with a total area of 300 sq km. Here the pollution is experienced at the highest levels in the country. Being the central hub, people from all regions of Bangladesh come to Dhaka, making it known in the world as one of the citied with the most dense population, with a density of 23,234 people per sq km [63]. As of 2022, the population in the metro area of Dhaka is about 22.5 million, which is 3.39 % more than that of 2021 [62]. Air pollution has clutched this city with a calamitous grip with no sign of subsiding, mainly caused by the increasing vehicular and machinery emissions, huge accumulation of dust particles, industrial fumes and brick kiln releases that contain high amounts of toxic soot. In this city SO_x, NO_x, and Exhaust emissions are 0.3 tonne per hour, 0.8 tonne per hour, and 13.5 tonne per hour, respectively [5]. These are the sources of polluting components in the air like nitrogen dioxide (NO_2), sulfur dioxide (SO_2), ozone (O_3) and carbon monoxide the presence of other hazardous pollutants, volatile organic compounds (VOC's) like formaldehyde, methylene chloride and benzene, and also black carbon [59]. These pollutants have destructive effects on the health of the population. The contaminated zone of SO_2, defined as an average concentration of more than 40 parts per billion, stretched from southwest to northeast, as well as across to the Buriganga river in Dhaka [3].

In order to overcome these issues, detailed monitoring of the pollution in different areas and controlling the sources of pollutants have become a must for Bangladesh. The critical need is not only to minimize air pollution, but also to implement systems, equipment, and computer networks to keep a careful eye on fast polluted air [4] [6]. Monitoring the pollutant levels of different areas of the capital, collecting more precise and accurate data to calculate the AQI of specific areas and predicting the pollution by the use of IoT (Internet of Things) architecture and machine learning is what the aim is in this research. The goal is to make the people aware of the pollution in the various populated areas of the city by providing access to data of air components that is narrowed down to specific zones.

1.1 Motivation

The average life expectancy of a person living in Bangladesh has been reduced by 2.91 years due to the air pollution [61]. Various respiratory diseases and complications and also impediments on other body systems are caused by air pollution. Respiratory ailments are common among the people living Bangladesh. Pollutants in the air cause diseases like pneumonia and also lung cancer. Also, since different kinds of particles and pollutants are being breathed in, they affect other body parts like the heart, reproductive organs and such [26]. Around 50,000 child deaths due to pneumonia occur in Bangladesh every year. In 2021, the chief hospitals for respiratory diseases treated around 210,000 patients, from which 60,000 were admitted into the hospitals. This number of patients is a lot more than that of 2020 and especially in the dry season, there is a 25% rise seen in the people affected by lung infection/diseases [64]. Because of environmental pollution and other related causes, about 234,000 deaths were caused in 2015, of which 80,000 were from the urban areas in Bangladesh. Around 21% of the total deaths in the country are caused by the environmental hazards. In 2015, beside the loss of life, there were also economic costs of labor because of the deaths and diseases caused by air pollution, valued at $1.4 billion in the urban areas and $310 million solitarily in Dhaka City. This equals 0.6% and 0.1% of Bangladesh's GDP in 2015 [30]. It is solely because of these reasons that is what motivated to find a way to seclude different zones, accurately monitor the air components of those areas to measure the AQI and produce a precise result using IoT devices. The data is used to make an AQI prediction model through machine learning. This data is to be made easily accessible to the public so that one may employ it as they want.

1.2 Contribution

Clean Air Sustainable Environment (CASE) is a project that the Ministry of Environment and Forestry of the Government of the People's Republic of Bangladesh has undertaken to adopt Sustainable Environmental Initiatives (SEI) in the main sources of pollution in the country, that is, emissions from vehicles and brick kilns. In the CASE website [9], one may find access to daily air component data from the reports of past years. This daily data is presented in the form of tables classified by months and years but it is incomplete and inconsistent. There is also a sizable amount of missing data. After having contacted the website, it was found that one has to pay a substantial amount of money to gain access to the complete data from the website. Similar is the case with another website called Gspatial, which asks for an even larger sum of money in order to give access rights to the data to public users [1]. Another website, namely Dhaka US Consulate Air Pollution: Real-time Air Quality Index (AQI) has only the P$M_{2.5}$ component data, which is also very inconsistent and has missing data [65]. Moreover, these data are the daily averages only and have no hourly data present.

This research's idea is to collect environmental data more consistently and frequently, and to make it presentable and easily accessible for public use. This is to be done using IoT device. The device is made using a microcontroller and necessary sensors that measure the air components of any area at a set frequency. This data is then stored and also used to build prediction models using machine learning, which is done by Multivariate Time Series Analysis.

1.3 Scope of the project

The internet of things, in short, IoT refers to any system of computers or computing devices, machines and other objects which are interrelated and are able to transfer data over a network autonomously without any sort of human interaction [58]. This sort of architecture is used as the basis of this research.

The device is made using a Raspberry Pi microcontroller and sensors for measuring the quantity of air components. It will continuously take records of the data that it collects and create data files and records. The records are made available for public use so that anyone willing to research about the pollution level or work on implementing any remedy or initiative for the reduction of air pollution may have access to the necessary information.

The collected data is also trained and tested using time series algorithms to create an AQI level prediction model. Different algorithms are used for this to meticulously conclude which is the most accurate in predicting the AQI level.

1.4 Outline

At the beginning in Chapter 1, an overview of the topic by introducing the different aspects of this research has been presented. After that, a detailed discussion of the theory behind this research is given in Chapter 2. Afterwards, in Chapter 3, a brief about research that has been conducted in the same field is documented. Chapter 4 contains specifics about the various areas of the investigation and research of this topic and also the tools that have been used. In the later Chapter 5, the employment of the system, execution of data collection and training using machine learning models has been explained. Then in Chapter 6 the results that have achieved by this research are discussed. Lastly, in Chapter 7 a final conclusive review of the project, suggestions and remarks are provided in the last chapter.

Chapter 2

Background

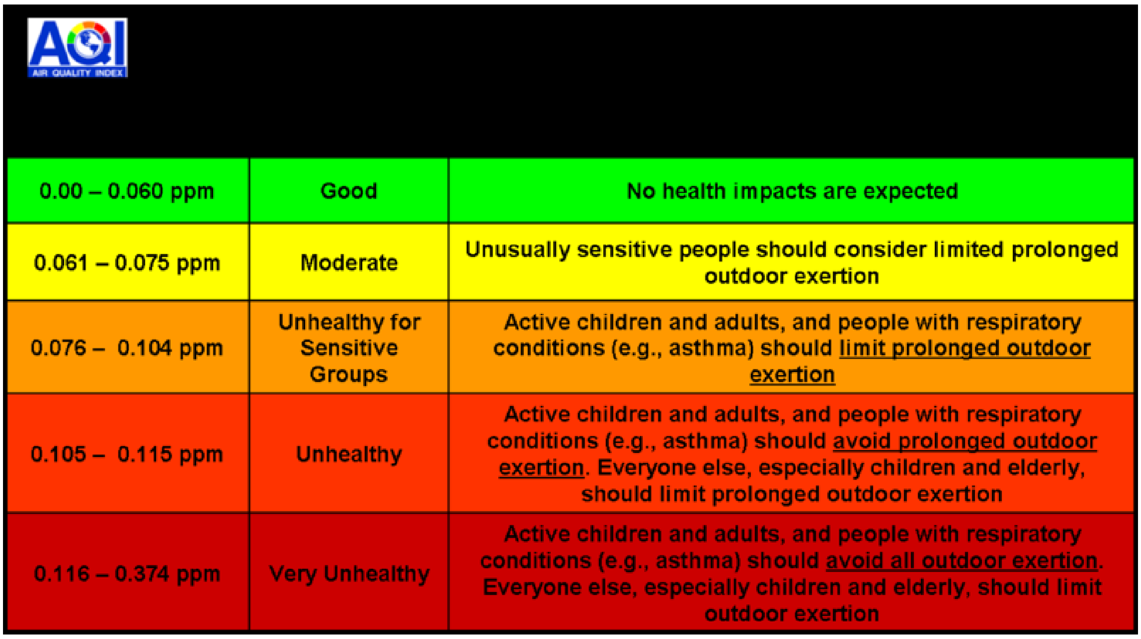
AQI (Air Quality Index) is a global standard of understanding how polluted the air of an area is. The concentration of various air components that are the main causes of air pollution is the basis of AQI calculation. Our research is based on the measurement of air components using IoT end devices and using machine learning to predict the concentrations of the pollutants.

2.1 AQI

The basis of this project is the measurement of the different air components that are required for determining AQI of an area. AQI, or Air Quality Index, is the basis of deciding the level of pollution of any area, depending on the amount of some harmful components that are present in the air above safe levels [1]. It is based on the AQI that decides whether the environment is healthy, unhealthy or unfit to live in.

AQI is basically a scale that may have a value of $0 - 500$. This value is in proportion to the amount of harmful components present in the air and it determines the health concerns related to the environment of any area.

A general breakdown of the different levels of health concerns centered on the AQI are shown:



0.00 – 0.060 ppm	Good	No health impacts are expected
0.061 – 0.075 ppm	Moderate	Unusually sensitive people should consider limited prolonged outdoor exertion
0.076 – 0.104 ppm	Unhealthy for Sensitive Groups	Active children and adults, and people with respiratory conditions (e.g., asthma) should limit prolonged outdoor exertion
0.105 – 0.115 ppm	Unhealthy	Active children and adults, and people with respiratory conditions (e.g., asthma) should avoid prolonged outdoor exertion. Everyone else, especially children and elderly, should limit prolonged outdoor exertion
0.116 – 0.374 ppm	Very Unhealthy	Active children and adults, and people with respiratory conditions (e.g., asthma) should avoid all outdoor exertion. Everyone else, especially children and elderly, should limit outdoor exertion

Figure 2.1: Standard Air quality index ratings [10]

US-EPA (US Environmental Protection Agency) AQI and Indian AQI, both having same ranges in the scale, are the two types of AQI that are used measure the air quality and pollution level [52]. Six main contaminants are used to measure the US AQI. Sarun [27] et. al. in their paper talked about these particulate matters or aerosols, that are, $PM_{2.5}$ and PM_{10} and compounds like Ozone (O_3), carbon monoxide (CO), sulfur dioxide (SO_2), and nitrogen dioxide (NO_2) are needed to calculate AQI.

The major pollutants present in the air, as mentioned before, have certain amounts of presence, above which it is unsafe to be exposed to those pollutants and have several kinds of damaging effects on human body. A brief about these components is given henceforth.

1. Particulate Matter/Aerosols ($PM_{2.5}$ and PM_{10})

Minute particles that are mixed with water droplets in the air form particulate matter. PM_{10} are those particles that have a diameter of 10 m. They are larger in size than $PM_{2.5}$, which are ultra-small particles having a diameter of 2.5 m [43]. Breathing in $PM_{2.5}$ is more hazardous than breathing in PM_{10} because $PM_{2.5}$ is highly probable to go and burrow into the deeper parts of the lungs, causing lung tissue damage and inflammation, irritation in respiratory organs, respiratory ailments like asthma, coughing, sneezing and such, whereas PM_{10} may only deposit on higher surfaces and airways of the lungs [56]. The sources of particulate matter are mainly from building and other constructions, demolitions, smoking and naturally due to volcanic eruptions and earthquakes. Safe level of P$M_{2.5}$ is 0 – 12 g/m3 and that of PM_{10} is 0 – 54 g/m3 according to US-EPA [52] [57].

2. Ozone (O_3)

The protective layer that exist within our atmosphere that blocks by absorbing harmful ultra-violet rays of the sun and prevents them from reaching the earth's surface is composed of ozone. Although ozone is necessary for this, ground ozone, that is, ozone gas on the surface layer of the atmosphere can be very harmful to human beings and plays a part in air pollution.

Ozone consists of three oxygen atoms and is produced by the chemical reaction of volatile organic compounds (VOC) and oxides of nitrogen (NO_x) when there is the presence of sunlight [48]. People breathing in ground ozone may develop breathing problems because it reduces the functions of lungs, causes inflammation of breathing routes and irritation in the throat, nose and eyes. Safe exposure level of ozone is 0 – 0.054 ppm according to US-EPA [52] [57].

3. Carbon Monoxide (CO)

Carbon monoxide is a chemical usually naturally existing in gaseous form in the air. It is released from incomplete combustion fuel within engines of motor vehicles, boats and other equipment. Burning of fuel in the presence of oxygen (O_2) produces carbon dioxide (CO_2) as a by-product, which is a greenhouse gas itself, but if the necessary amount of oxygen is not present while fuel is being burned, CO is produced as by-product.

CO is a dangerous chemical for the human body because it is reactive and creates unwanted substances by reacting with the necessary compounds present in our body, causing carbon monoxide poisoning. It interferes with oxygen-hemoglobin binding in the body, which is the main way of oxygen circulation in the body, and causing chest pain, diseases related to heart and blood, problems in vision and mental capacity. According to US-EPA, safe level of CO is 0 – 9.4 ppm [52] [57].

4. Nitrogen Dioxide (NO_2)

Nitrogen dioxide is another highly reactive gas that is present in the atmosphere. It is released into the atmosphere from vehicular emissions, combustion of fuel, electricity generation and industrial discharge. The harmful effects of nitrogen dioxide are similar to that of carbon monoxide since they are both reactive gases which changes other chemicals it comes in contact with into harmful chemicals. It is a main component of smog. When inhaled, nitrogen dioxide may cause critical damage to the lungs and the heart. Its effects are coughing and wheezing, inhibited lung function, inflammation of airways and asthma which can be so grave at times that the effected person needs to be admitted to a hospital [35]. The safe level of nitrogen dioxide in the air is 0 – 53 ppb [52] [57].

5. Sulfur Dioxide (SO_2)

Sulfur dioxide is a gas which is acidic in nature. It is colorless and has a burning odor. This gas is corrosive and also reactive in nature, combining with other compounds in the atmosphere, forming sulfuric acid and sulfates. Vehicular emissions, fuel combustion, electricity generation and industrial releases are the main sources of sulfur dioxide in the atmosphere. It produces sulfuric acid and it is one of the major causes of acid rain that damages plants and also buildings and other man-made structures. The harmful effects of sulfur dioxide in the atmosphere similar to those of nitrogen dioxide, causing coughing and asthma, lung function inhibition and shortness of breath [36]. Its safe level according US-EPA 0 – 75 ppb [52] [57].

2.2 Calculation of AQI

Air Quality Index has a value from 0 to 500 and using this scale is how one may comprehend the contamination of the air by the different kinds of pollutants. AQI is calculated individually using the air pollutant's concentration that exists of an area. The worst value of AQI, that is, the highest value that is calculated is the AQI for that location.

Every pollutant may not be used to measure the AQI. In that case, concentrations of at least three pollutants should be considered, one of which should necessarily be P$M_{2.5}$. If this data is not available, then it is considered to be insufficient for the calculation of AQI. The concentrations of each pollutant are either 1 – hour, 8 – hour or 24 – hour averages. At least 18 hour averages of concentration are required to get a daily average value of concentration, that is, 24 – hour average.
The calculation of AQI is accomplished by following the aforementioned steps:

1. The values of concentration of each pollutant are to be rounded up accordingly:
-Concentration of ozone is rounded up to 3 decimal places
-Concentrations of P$M_{2.5}$ and CO are rounded up to 1 decimal place
-Concentrations of SO_2 and NO_2 are rounded up to nearest integers

2. AQI is calculated using the given equation for each pollutant individually. Then the highest index rounded up to the nearest integer is the required value of AQI. The formula:

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}}(C_p - BP_{Lo}) + I_{Lo}$$

Here, C_p = truncated concentration of p pollutant
I_p = index for p pollutant
BP_{Hi} = breakpoint of concentration which is greater than or equal to C_p according to Table 2.1
BP_{Lo} = breakpoint of concentration which is less than or equal to C_p according to Table 2.1

I_{Hi} = Value of AQI that corresponds to BP_{Hi} according to 2.1
I_{Lo} = Value of AQI that corresponds to BP_{Lo} according to 2.1
3. The breakpoints that contain the concentrations in their range are to be found out from the given table:

O3 (ppm) 8-hour	O3 (ppm) 1-*hour*[1]	$PM_{2.5}$ (μg/m^3) 24-Hour	PM_{10} (μg/m^3) 24-Hour	CO (ppm) 8-hour	SO_2 (ppb) 1-hour	NO_2 (ppb) 1-hour	AQI	
0.000-0.054	-	0.0-12.0	0-54	0.0-4.4	0-35	0-53	0-50	Good
0.055-0.070	-	12.1-35.4	55-154	4.5-9.4	36-75	54-100	51-100	Moderate
0.071-0.085	0.125-0.164	35.5-55.4	155-254	9.5-12.4	76-185	101-360	101-150	Unhealthy for sensitive Groups
0.086-0.105	0.165-0.204	(55.5–150.4)[3]	255-354	12.5-15.4	(186 – 304)[4]	361-649	151-200	Unhealthy
0.106-0.200	0.205-0.404	(150.5–250.4)[3]	355-424	15.5-30.4	(305 – 604)[4]	650-1249	201-300	Very Unhealthy
([2])	0.405-0.504	(250.5–350.4)[3]	425-504	30.5-40.4	(605 – 804)[4]	1250-1649	301-400	Hazardous
([2])	0.505-0.604	(350.5–500.4)[3]	505-604	40.5-50.4	(805 – 1004)[4]	1650-2049	401-500	Hazardous

Table 2.1: Break Points for the AQI [21]

2.3 AQI Prediction by Machine Learning

After having collected the concentrations of different air components, the next step of this particular research is to train that data using machine learning to predict AQI data.

1. Machine Learning

Holzinger [29] et. al. in their paper has discussed about how the popularity of statistical machine learning (ML) approaches has boosted interest in Artificial Intelligence (AI). ML, or machine learning, is a subject matter related to artificial intelligence, that is, AI. It is actually a subclass of AI. ML is a way of “studying” data by computers or systems and automating the process of analytical model building. The idea of automation is based on the fact that a system can make decisions or take an action without or with minimal interaction of humans by “learning” from a large amount of data and finding out the patterns in that data [1].

A system trained by ML analyzes statistical data from available databases so that it can make assessments according to how it has been trained and also be able to predict data.

The ultimate goal of IoT is to bring electric technology that provides terminals with simplicity of use, wireless access control, and flexibility [42]. This research's system is built using an IoT system that can measure the various pollutants present in the air that are required for the measurement of AQI and record that data, creating a data set. The data set is then trained using machine learning algorithms which will allow the system to predict the AQI based on the investigation of ready data.

2. Prediction algorithms

The data that is collected by this study's system, is ready to be modeled in accordance to multivariate time series forecasting. There are more than one time-dependent variables in multivariate time series, which do not depend only on past values but also has dependency on other variable/variables in one or the other [2].

Regression analysis is used to identify the data patterns in the data set. Various regression algorithms, that are, Linear Regression, Random Forest Regression, KNN (K-nearest neighbors) algorithm and Lasso Regression are employed and made use of to find the interred patterns existing within our data. These algorithms were chosen because various research works show that these are suitable for time series forecasting. Along with these regression algorithms, a deep learning algorithm LSTM is employed as well.

2.4 IoT System Development

IoT architecture implies a system of computers, devices and other peripherals connected together in a network that are able to transmit data among them. The Internet of Things (IoT) is a global network of smart solutions that can perceive and link to their environment, as well as communicate with people and other services [31]. The basis of this research is the IoT system that have been built using a device made of a certain few components that will allow to read and collect the air component data, store that data, create a data set, make the data accessible for the public to use as they wish, transfer data to other devices. This dataset helped this research to train, test and predict the amount of air components in the future.

One of the main components of this study's device is a microcontroller (Raspberry Pi and Arduino Uno). With the microcontroller, there will be a few sensors connected which will read the amount of different components/pollutants present in the air. The sensors are PMS5003, MQ – 7, MQ – 131, MQ – 135 to measure the amount of $\mathrm{P}M_{2.5}$, CO, $\mathrm{N}O_2$, and O_3 respectively and DHT – 11 for the measurement of temperature and humidity of the environment. These sensors read the amount of corresponding components in the air and collect that data to store it. A Micro SD Card is also connected to the microcontroller to store the collected data.

Software such as VNC Viewer and VNC server are used to remotely control the device and Node RED is used for GUI-based programming.

Chapter 3

Related Works

This chapter explains the recent works of AQI prediction and data collection for time series forecasting. In several ways, data collection, as well as various methodologies and models, are employed in order to achieve the highest accuracy for the time series forecasting.

3.1 IoT-based

For IoT-based data collecting and monitoring, Zheng [18] et. al. designed and implemented an IoT system. This system had monitoring nodes that had a Microcontroller Unit (MCU), and portable sensors, and the nodes were battery-powered to collect information about the quality of the air and it's components, which is then transferred by a Machine-to-Machine (M2M) communication technique, Low Power Wide Area technique (LPWA). They adopted a three-tier (sensing layer, network layer, and application layer) hierarchical IoT architecture. The hardware system's monitoring node collected information about the air components in real-time and this monitoring node consisted of four modules in total (Controller module, Sensor module, Power module, LPWA Tx Module). The monitoring nodes are powered by lithium batteries so these lithium-powered batteries need to be charged, which can be done using a solar panel. The negative side of this IoT system is the solar panel's output voltage is often so large and variable to charge the battery.

On the other hand, Agarwal [16] et. al. in their paper propounded a framework to monitor the parameters of the city environment. In the given framework they used a system of low power and low-cost minicomputer, Raspberry Pi. From this paper temperature, humidity, pressure, CO these are measured but they gave no emphasis on the particulate matter and that being so the city environment is left incomplete as one of the prominent components in the air that contributes to pollution. In another study conducted by Chhikara [45] et. al. in their paper talked about aerial sensing and how they have used a fleet of UAVs to collect air quality data by using built-in sensors. UAV or the Unmanned Aerial Vehicle has the ability to collect the air quality data with temporal resolutions and high spatial. Here the UAV swarm can keep track of time and as sensors need to be calibrated so commands to recalibrate are sent once in two weeks. As they have used a swarm of UAVs and the number of connections is n*(n-1)/2, hence a substantial amount of expense is created which will hinder the delivery of AQI monitoring in the long run. Another method is proposed by Desai [19] et. al. where they used BeagleBone Black which is an open-source development platform for developers at a minimal cost. The BeagleBone board is embedded with gas sensors and this proposal acquired Carbon Dioxide and Carbon Monoxide levels in the air. Beagle bone's GPIO pins and the integrated ADC can be configured and enabled using Python programming. Sockets are typically used to connect two platforms, such as cloud and python, together. The data from the sensor is uploaded to the Azure Cloud using Python SQL.

Then again Moses [40] et. al. made an effort to create a cost effective wireless system for the monitoring of air pollutants. The device monitors pollutant levels and updates Google Maps accordingly. A less polluted region might assist people to redirect their trips. Here they have used Narrowband IoT (NB-IoT) which is based on a 4G LTE network that provides long battery life and lower cost. The NB-IoT nodes can integrate with cloud services on their own hence they transmit the estimated sensor values to the network. Korunoski [32] et. al. in their paper explained that in order to keep an eye on urban air pollution levels and implement measures to reduce them, intelligent architecture for pollution prediction and monitoring must be developed. The sequential modeled architecture begins with building an architecture for determining the pollutant and its amount, and the system is constructed by creating a method imputed polluted field for the recorded pollutant. Then, a deep learning model with long-term memory may be used to anticipate the updating of contaminated data. As a result, the server is able to make a forecast for the future and deliver data on alarming levels.

3.2 Air Components

Now for different models different air components were considered. For the examination of air quality data near highways, Akula [8] et. al. presented a dispersion model which was used to find pollutants emitted from vehicles and specifically Nitrogen Oxide (NO) emission. Wind direction, temperature, wind speed and humidity were considered as meteorological parameters. Then, Dogruparmak [13] et. al. in his paper considered SO_2, NO_2, NO, PM_{10}, NO_x, O3 as air components in order to monitor air quality. The air monitoring system is distributed in stations and hence both PM_{10} and SO_2 components are measured in all of the stations, they are both considered for analysis. Modali [20] et. al. in his AQI forecasting research models considered SO_2, O_3, NO_2, CO, P$M_{2.5}$ and PM_{10} air components. Another research regarding AQI and the impact of meteorological conditions on AQI was done by Qiao [53] et. al. where they considered wind speed, humidity, temperature and rainfall as the factors of meteorology. They also collected pollutants such as P$M_{2.5}$, NO_2, O_3, PM_{10}, CO and SO_2 from the areas of Taipei city as their research was based on this city's air components.

3.3 Models

Various methods were used in order to forecast AQI in different researches. Using Machine Learning approaches, the paper of Urda [39] et.al. presented a two-step forecasting approach for obtaining future AQI values eight hours ahead of time. In the paper they stated that modeling non-linear time series is possible using ANN, SVR, and LSTM, and they may be trained to generalize accurately when a new database is supplied. Then again, Zhao [44] et.al. proposed a model in terms of both spatial dimensions and temporal dimensions that predicts AQI in a non-monitoring area. It is a first-based temporal dimension model, which uses an improved K-Nearest Neighbor (KNN) algorithm in order to forecast AQI values among monitoring stations, with a 92 % acceptability rate for one-hour prediction.

Regarding regression and classification Kumar [7] et.al. in their research paper, which is about the forecasting of stock index movement, showed comparison between Random Forest Regression and SVM. In this study, they stated that Random Forest outperforms Linear Discriminant Analysis, Neural Networks and Logistic Regression. Along with the traditional methods, deep learning was also applied in few researches. Such a research was conducted by Zhang [37] et. al. regarding time series forecasting of the spread of COVID-19 in Canada using the LSTM networks. Long short-term memory (LSTM) networks, a deep learning approach for forecasting cases of COVID-19 in the future,had been mentioned in the research. So the LSTM network was a great help in the crucial time of COVID-19 as well. This paper shows the successful prediction that, at around June 2020 the cessation of the outbreak would take place. This was derived from the results of the Long short-term memory (LSTM) networks. LSTM network was also mentioned in a paper by Hansson [21], where he predicted stock return with LSTM networks. In this research, recurrent neural networks and LSTM (long short-term memory) are used so that financial time series forecasting can be performed on the return data of three stock indices.

Another study conducted by Mussumeci [41] et.al., where they compared Random Forest, Lasso and LSTM regression for the forecasting of dengue epidemic by machine learning. It has been challenging to obtain such projections using traditional time series models. For a weekly anticipation of dengue in 790 Brazilian cities, they presented and compared machine learning models like Random Forest regression and LASSO with the LSTM network. For capturing the spatial component of the transmission of disease, they have employed multivariate time series as predictors, as well as utilized time series from cities with similar cases. The LSTM recurrent neural network model outperformed with the fewest predictive errors among the compared models in predicting dengue outbreaks.

Nonetheless, there are many more techniques available for time series forecasting along with the mentioned research.

Chapter 4

System Methodology

This is a system of IoT devices which are used to collect the necessary data for the research. That data is then gathered into a dataset which is required for the prediction thorugh machine learning.

4.1 System Architecture

The flowchart shown below is the proposed architecture of this study.

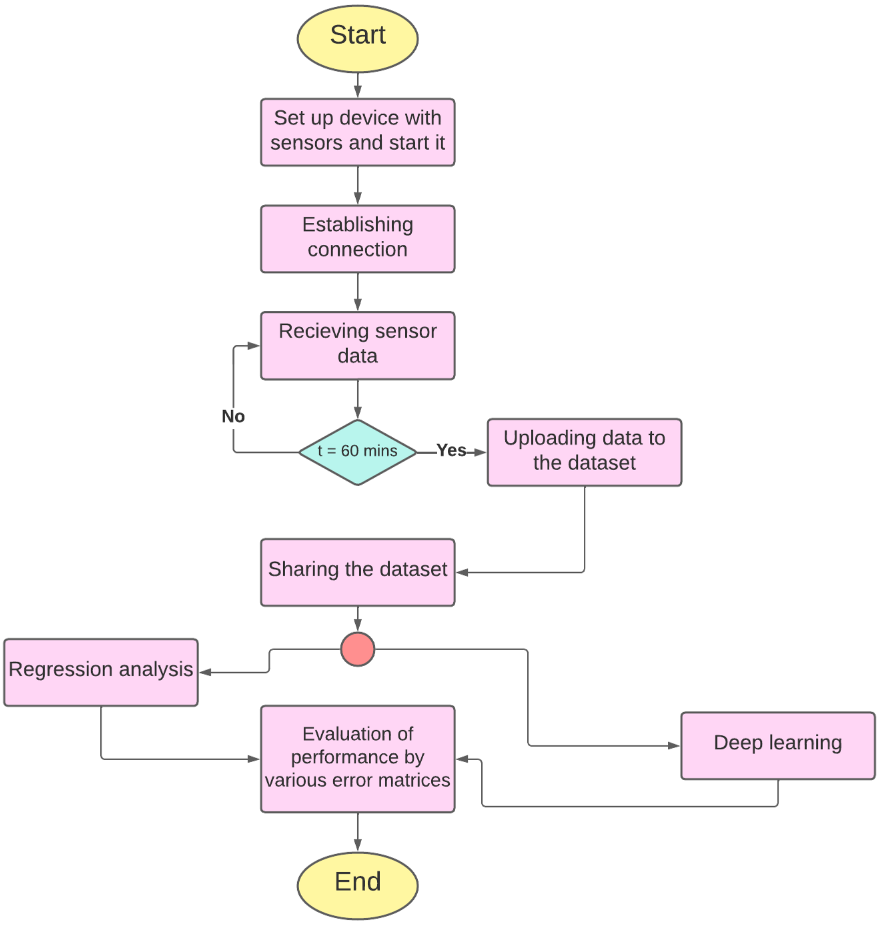

Start
Set up device with sensors and start it
Establishing connection
Recieving sensor data
No
t = 60 mins
Yes
Uploading data to the dataset
Sharing the dataset
Regression analysis
Deep learning
Evaluation of performance by various error matrices
End

Figure 4.1: Overview of The System

4.2 Hardware

The end device that is prepared for the research is based on IoT architecture. Hence, it needs hardware components.

4.2.1 Components

1. Raspberry Pi Model 3 B with Power Adapter
2. Arduino Uno
3. Breadboard
4. PMS5003 By Plantower
5. MQ-7 (Carbon Monoxide Sensor)
6. MQ-131 (Ozone Sensor)
7. MQ-135 (NO_x Sensor)
8. DHT-11(Temperature and Humidity Sensor)
9. 64GB Micro SD Card
10. Real-Time Clock Module (DS3231)

1. Raspberry Pi Model 3 B with Power Adapter

The third generation Raspberry Pi is the Raspberry Pi 3 Model B. This robust bank-card-sized single board microprocessor may be utilized for a variety of functions. It also has a wireless LAN and Bluetooth connection, making it an excellent choice for strong networked designs. It is ten times quicker than the first generation of Raspberry Pi.

On our desktop PC, we can do all of the tasks we expect. The properties of this little device make it a powerful IoT device. We selected Raspberry Pi since we are working on an IoT project. It has Wi-Fi functionality, which allows us to store and manage all the data on the cloud. Concerning the electrical supply, the Raspberry Pi 3 can be operated for a long time using a regular Mobile Power Bank that we use to charge our Android phone every day. Data can also be stored on the onboard removable storage of it.

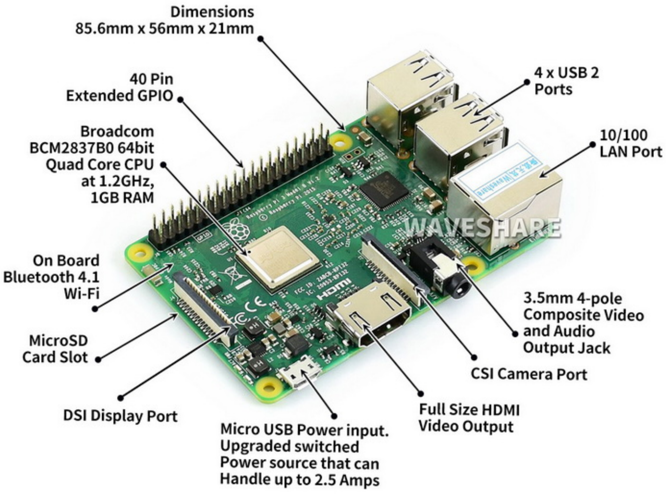


Figure 4.2: Components of Raspberry Pi 3 Model B [60]

Raspberry Pi 3 Model B Hardware Specifications:

1. Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU
2. 1GB RAM
3. ARMv8 is the CPU architecture
4. On-board BCM43438 wireless LAN and Bluetooth Low Energy (BLE)
5. 10/100-Base-T Ethernet Port (RJ-45)
6. 40 pins extended GPIO are available for connecting and operating various
7. electronic devices and Raspberry Pi modules
8. 4 USB 2 ports
9. 3.5mm connection for 4 pole stereo audio and video output
10. Full-size HDMI
11. A microSD card slot on the motherboard
12. Micro SD slot for installing and storing the operational information assets
13. Raspberry Pi Camera can be connected to the CSI camera port
14. A DSI display port is used to connect a Raspberry Pi touchscreen display
15. Switched Micro USB power supply upgraded to 2.5A [1]

Furthermore, it now includes built-in WiFi and Bluetooth connection as well as better power management to handle more powerful external USB devices. A 2.5A converter is suggested to fully utilize the Raspberry Pi 3's better power management and to handle even more powerful devices via the USB ports.

2. Arduino Uno

The Arduino Uno is an open microcomputer designed by Arduino.cc that is based on the Microchip ATmega328P microcontroller. Uno is an 8-bit ATmega328P microcontroller-based microcontroller board. To enable the microprocessor, it contains additional elements such as a crystal oscillator, serial ports, a power amplifier, and so on. Arduino is written in the C/C++ programming language.

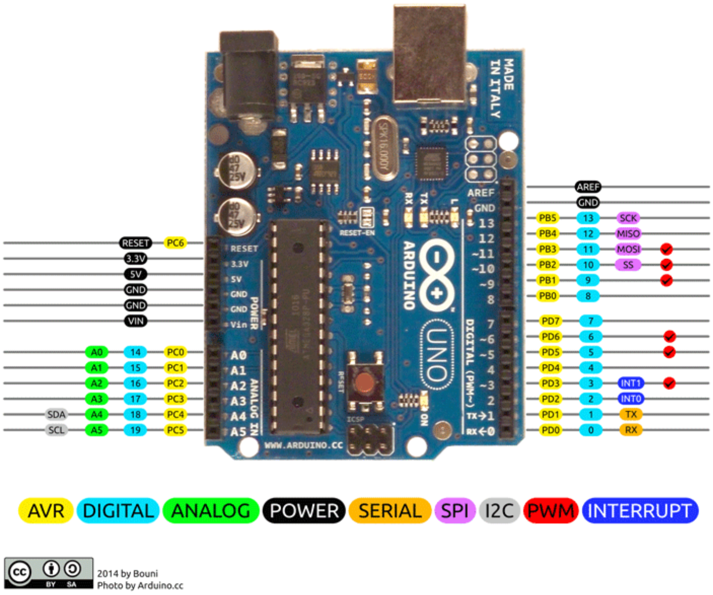


Figure 4.3: Pinout figure of the Arduino UNO [46]

The technical specifications of the Arduino Uno are listed below:

1. Operating voltage of 5 volts
2. The suggested input voltage is 7-12V, however, the maximum input voltage is 6-20V
3. 14 digital I/O pins, 6 of which output PWM
4. DC Current per I/O Pin: 20 mA
5. DC Current for 3.3V Pin: 50 mA
6. Flash storage of 32 KB (ATmega328P), 0.5 KB of which is consumed by the bootloader
7. SRAM: 2 KB (ATmega328P)
8. EEPROM: 1 KB (ATmega328P)
9. Power sources include a DC power jack and a USB port.
10. Clock Speed: 16 MHz

Furthermore, because it is an open-source platform, the boards and software are easily accessible, and anybody may alter and modify the boards for improved functionality. The software used for Arduino devices is known as IDE (Integrated Development Environment), and it is free to use. It does, however, need some fundamental skills to master. However, the Arduino UNO can only run one uploaded code in a loop at a time.

Pin Class	**Pin Name**	**Details**
Power	Vin, 3.3V, 5V, GND	Vin: This is the power source of arduino uno. 5V: Can be used as output to power several sensors and other components. 3.3V: Power supply controlled to 3.3V via on-board voltage regulator. 50mA current draw at max. GND: Ground pins to be used in sensors and other components.
Reset	Reset	A soft reset button for the arduino.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Table 4.1: Arduino Pinout Connections [46]

3. Breadboard

A breadboard, often known as a protoboard, is a building platform for electrical design. The breadboard has numerous holes that allow one to effortlessly insert electrical components into the prototype, which means designing and testing an early version of an electronic circuit. As no welding is required, changing connections and replacing electronics is simple. The components are not harmed and may be reused.

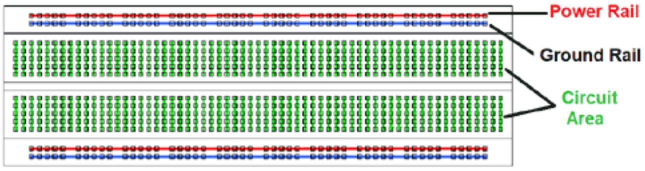


Figure 4.4: Pinout of Breadboard Connections [24]

The breadboard's vertical columns are known as terminals, while the horizontal long rows are known as power rails since they are primarily used to connect the power source to the breadboard. Red lines show positive rails, whereas negative rails are denoted by black lines. The component of the circuit that connects directly to the negative terminal of the power supply or battery is known as the ground. The breadboard's attributes include the following:

1. The pitch or hole size is 2.54mm
2. There are two Distribution Strips
3. The wire size ranges from 21 to 26 AWG
4. There are 200 tie points
5. 630 are the tie points inside IC
6. The maximum operating voltage is 1,000V AC
7. DC500V or 500MΩ is the insulation resistance
8. The power is 5Amps
9. ABS plastic with a color legend and its heat distortion is 183° F (84° C)
10. The dimensions are 6.5*4.4*0.3 inch

To be specific, The breadboard contains metal strips below it that link the holes on the top of the board. The top and bottom rows of holes are joined horizontally and split in the middle, whilst the remaining holes are joined vertically.

4. PMS5003 By Plantower

The PMS5003 is a digital and universal material density sensor that may be used to determine the number of scattered particles in the air. The Plantower PMS5003 laser dusting analyzer. The sensor measures the value of dust particles floating in the air using the laser light scattering method. The sensor delivers an accurate and dependable readout of the P$M_{2.5}$ value. This sort of P$M_{2.5}$ sensor does not require external calibration; instead, it calibrates from within the system.

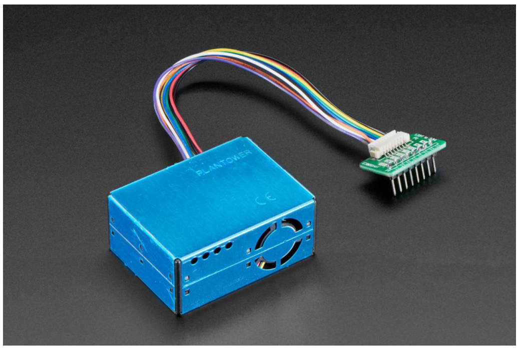


Figure 4.5: PMS5003 By Plantower Air Quality Sensor [66]

Configurations of Plantower PMS5003:

1. Voltage range: 4.5 V to 5.5 V
2. There are two Distribution Strips
3. Work power consumption is less than 100 mA, and standby power consumption is less than 200 A
4. Temperature range: -10 to 60 degrees Celsius
5. 1 g/m^3 resolution
6. 50 percent sensitivity - 0.3 m and Humidity (work): 0-99 percent and 98 percent - 0.5 m and higher

This sensor can be used in changeable sensors to measure the percentage of suspended particles in the atmosphere or other environmental improvement devices to offer accurate percentage data in real-time.

Pin	Function	Description
1	VCC	Supply voltage 5V
2	GND	Ground
3	SET	HIGH or SUSPENDED – work mode LOW – sleep mode
4	RXD	UART/TTL data recieve
5	TXD	UART/TTL data transmit
6	Reset	LOW to reset
7	NC	Not Connected
8	NC	Not Connected

Table 4.2: PMS5003 Pinout Connections [66]

5. MQ-7 (Carbon Monoxide Sensor)

This is a basic Carbon Monoxide (CO) sensor for measuring CO levels in the atmosphere. The MQ-7 is sensitive enough to detect CO-gas concentrations ranging from 10 to 500ppm. The simple analog voltage interaction of the sensor utilizes only one analog input pin from the microprocessor. This gadget has digital and analog outputs but must be calibrated manually. The digital output threshold level may be simply modified using the board's standard. The MQ-7 sensor module is conveniently interconnected with microcontrollers, such as Arduino.

Figure 4.6: Carbon Monoxide Sensor - MQ-7 [70]

Features of the MQ-7 (Carbon Monoxide sensor):

1. DC 5 V is the operating voltage
2. Carbon monoxide monitoring with hypersensitivity
3. The analog output voltage increases as intensity increases

This CO gas sensor detects CO concentrations in the air and outputs the information as an analog voltage. The sensor works at temperatures ranging from -10 to 50°C and draws below 150 mA at 5 V. It is effective, with long service life and stable performance, and characteristics of rapid response and recovery.

Pin No.	**Symbol**	**Descriptions**
1	DOUT	Digital output
2	AOUT	Analog output
3	GND	Power ground
4	VCC	Positive power supply (2.5V-5.0V)

Table 4.3: MQ-7 Pin Configuration [67]

6. MQ-131 (Ozone Sensor)

The MQ131 ozone sensor's gas detecting substance is a semiconducting metal oxide with strong electrical properties in clean air. When ozone is present in the atmosphere where the sensor is placed, the sensor's conductivity reduces as the quantity of ozone gas in the air increases. A simple circuit may translate the change in conductivity to an output signal matching the gas concentration but it has to be physically calibrated.

Figure 4.7: Ozone Sensor MQ-131 Ozone Gas Detection Module [50]

MQ-131(Ozone Sensor) product information:

1. Working voltage: DC 5V
2. With an indicator light for signal output
3. LM393 main chip
4. TTL output effective signal is low level; when the output is low level, the signal light flashes and may be linked to the IO port of the mono microprocessor.
5. TTL level output and analog output with dual signal output
6. The analog output rises as levels increased, percentage Y is high, and voltage Y is high

This has a long lifespan and dependable KE stability. The Characteristics of it are rapid response and recuperation. With mounting holes for permanent installation, it may be connected and removed, making examination easier.

Pin Number	**Pin Name**	**Description**
1	Vcc	The pin is there to power the module, requires 5V
2	Ground	To connect the module to the system's common ground
3	Digital Out	Digital output pin, the Threshold value can be set by using the potentiometer
4	Analog Out	Analog output pin. Analog voltage based on the concentration of the gas

Table 4.4: MQ-131 Pin Configuration [50]

7. MQ-135 (NO_x Sensor)

This MQ-135 Gas sensor, like the other MQ series gas sensors, has a digital and analog output pin. The sensor can also detect gases such as ammonia (NH3), sulfur (S), benzene (C6H6), and CO2, but mostly NO_x and other dangerous and harmful gases and smoke. When the concentration of these gases in the air exceeds a registered total, the digital pin rises high. This threshold value may be altered using the onboard regulator. The analog output pin generates an analog signal that may be used to estimate the number of various gases in the environment.

Figure 4.8: Air Component Detection Sensor MQ-135 (NO_x Sensor) [68]

MQ-135 Gas Sensor Technical Specifications:

1. 2.5V to 5.0V operating voltage
2. 150mA power consumption
3. 5V is the average work voltage
4. 0-5V analog output at 5V Vcc
5. TTL Logic Digital Output: 0V to 5V at 5V Vcc

It has a wide spectrum of susceptibility to hazardous gases. This sensor's sensitive substance is SnO_2, whose resistance is lower in clean air. The gadget has digital and analog outputs but the output should be corrected by manual. The MQ135 Air Quality Sensor module is effortlessly interfaced with embedded systems, Arduino boards, Raspberry Pi, and other devices.

Pin Number	**Pin Name**	**Description**
1	Vcc	The pin is there to power the module, requires 5V
2	Ground	To connect the module to the system's common ground
3	Digital Out	Digital output pin, the Threshold value can be set by using the potentiometer
4	Analog Out	Analog output pin. Analog voltage based on the concentration of the gas

Table 4.5: MQ-135 Pin Configuration [51]

8. DHT11(Temperature and Humidity Sensor)

The DHT11 is a simple and inexpensive electronic thermometer and humidity sensor. It measures the airflow with a capacitive humidity sensor and a thermostat and outputs a digital signal on the data port, with no analog input pins required. It is quite simple to operate, however, data collection requires precise timing.

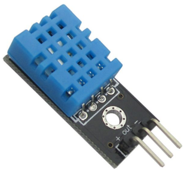


Figure 4.9: Temperature and Humidity Sensor: DHT11 [47]

DHT11 Specifications:

1. I/O and power from 3 to 5V
2. The maximum current used during conversion when requesting data is 2.5mA
3. Temperature measurements from 0 to 50 degrees Celsius are accurate to +2 degrees Celsius
4. With a 5 percent accuracy, it is suitable for reading humidity levels ranging from 20 to 80 percent
5. Once per second, a sampling rate of no more than 1 Hz is used
6. Compliant with RoHS
7. It has a 4.7K or 10K resistor that may be used as a pull up from the data pin to VCC
8. 4 pins separated by 0.1”

The DHT11 is a popular temperature and humidity sensor that includes a specific NTC to detect temperature and an 8-bit microprocessor to output temperature and humidity measurements as serial data. If anyone wants to connect DHT11 to Arduino, there are ready-made libraries that will help to get started quickly but the temperature and humidity must be manually calibrated.

Pin No.	**Pin Name**	**Description**
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

Table 4.6: DHT11 Pin Configuration [47]

9. 64GB Micro SD Card

64 GB micro SD cards are compatible with the newest speed class, with great random read/write speed, making them ideal for obtaining small chunks of data from random locations and shortening app launch time.

Figure 4.10: 64GB Micro SD Card [69]

This SD card was chosen since:
1. The functioning of the Raspberry Pi must be fluid, seamless, and instantaneous.
2. Although CSV files require less data, we employed this large storage for stability.

10. Real-Time Clock Module (DS3231)

The Real-Time Clock Module (or DS3231) is a component that measures time, either in conjunction with or independently of its Arduino card, via its cell.

The Arduino card measures the time since the module was turned on in milliseconds. The module is sent fully constructed and ready to use, with a battery included.

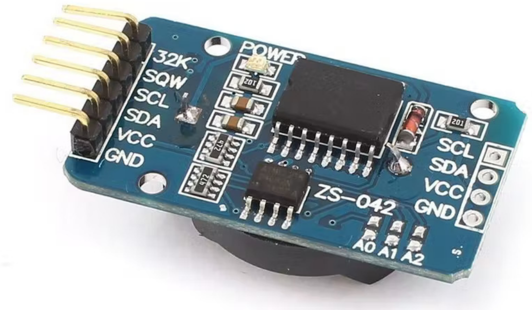


Figure 4.11: DS3231 RTC Chip [33]

The RTC's battery lasts a long period and provides accurate time when set. It has not encountered any technical issues in a long time. Also, because we are using LSTM, a timestamp is required, thus we are utilizing this module.

The DS3231 is a six-terminal device, two of which are not required to be used.

So we usually have four pins. These four pins are provided on the other side of the module with the same name.

Pin Name	Description
VCC	Connected to the positive of the power source.
GND	Connected to ground.
SDA	Serial Data pin (I2C interface)
SCL	Serial Clock pin (I2C interface)
SQW	Square Wave output pin
32K	32K oscillator output

Table 4.7: DS3231 RTC Pin Configuration [25]

4.3 Models

Regression analysis is performed to forecast the value of the dependent variable for individuals who have some knowledge about the explanatory factors, as well as to evaluate the influence of an explanatory variable on the dependent variable, which was required in this research. We also require arbitrarily complicated mappings from inputs to outputs and support for numerous inputs and outputs in our work since we are using data from several air components that might vary randomly; such structure can be learned using Deep Learning.

4.3.1 Regression Analysis

When it comes to statistics and machine learning, regression is one of the most well-known and well-understood algorithms. Regression analysis is a technique for identifying data patterns. It is a quantitative approach used to investigate and interpret the relationship underlying two or more variables of interest. The regression analysis approach assists in determining which aspects are essential, which may be discarded, and how they interact with one another. It is based on data modeling and comprises identifying the ideally appropriate system that goes through all data points while having the shortest distance between the connection and each data point.

1. Linear Regression

The most renowned and well-understood statistics and machine learning algorithms are linear regression. A simple linear regression model attempted to explain the relationship between two variables by using the best suitable straight line, known as a regression line. Basically, a line would be drawn, and then for each data point, the vertical distance between the point and the line can be measured and added up. The regression line will be the one with the smallest total of distances. When the sample size is small or the signal is poor, linear regression generally provides a decent approximation to the underlying regression function [11].
The generic single-equation linear regression model, maybe expressed such as:

$$Y = a + \sum_{i=1}^{k} b_i X_i + u$$

Here, Y is the dependent variable. The $X_1, X_2 ... X_i ... X_k$ is the k independent variables, a and bi are the regression coefficients expressing the model parameters for a certain sample, and u is a stochastic disturbance factor that can be represented as a result of the interaction of unidentified independent variables or indeed a randomly chosen element in the connection specifier [1]. The value of X, which is known as the independent variable, predicts the value of Y, which is known as the dependent variable.

2. LASSO Regression

LASSO is an acronym that stands for Least Absolute Shrinkage and Selection Operator. What it does is set the coefficient of characteristics that don't benefit the regression results enough to a very small value, practically zero. In both model prediction and variable selection, the Lasso approach has proven to be quite effective. Following Tibshirani's [2] groundbreaking work, an active research effort has been committed to this approach.
Below is the Lasso regression equation:

$$\sum_{i=1}^{n} (y_i - \sum_{j} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

This is equivalent to minimizing the sum of squares under constraint $\sum |\beta_j| \leq s$. Some of the βs are reduced to zero, resulting in a more interpretable regression model. λ is a tuning parameter, and when $\lambda = 0$, no parameters are removed. The estimate is the same as the one obtained using linear regression. As λ grows, more and more coefficients are set to zero and theoretically deleted; at $\lambda = \infty$, all coefficients are eliminated. Bias grows as λ increases and Variance grows as λ lowers.

3. Random Forest

Random forest is a Supervised Machine Learning Algorithm frequently utilized in Classification and Regression applications. Random Forrest generates multiple trees using recursive partitioning and then aggregates the results. Each tree is built individually using a bootstrap sample of the training data, which divides the parameter set into multiple parts based on one of the parameters and then repeats the procedure for each portion [17].

Random forest is a variety of machine learning models that provides predictions using an aggregation of decision trees. The more decision trees one implements with diverse conditions, the better a random forest will perform since it increases the forecasting accuracy. One of the key advantages of random forest is that it may assist in avoiding over fitting. The more decision trees is employed with diverse criteria, the better the random forest will perform because it is effectively boosting the prediction accuracy. This happens when the model begins to remember the data instead of attempting to generalize by generating predictions about future data. It essentially allows one to work around the limits of my data, which may not be totally representative of all users or all of the finest qualities in the model.

It can also contribute to the reduction of something else, called bias. Bias can emerge when a certain proportion of inconsistency is incorporated into the model. Bias happens when it does not divide the instance space appropriately while training. As a result, rather than seeing all of the data points, one may only see half of them depending on how one has set up the model.
When there are several decision trees or users, the Random Forest Algorithm may be used for such data. The random forest would utilize the outcomes of each decision tree to find an average connectivity or distance between all of the decision trees.

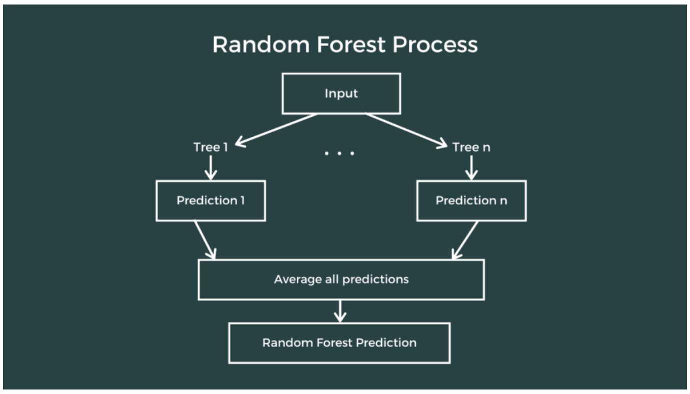


Figure 4.12: The random forest (RF) flowchart for regression [15]

Each tree is constructed by the algorithm using a distinct sample of input data. A fresh sample of characteristics is chosen for splitting at each node, and the trees operate simultaneously with no interaction. The forecasts from each tree are then combined to offer a common outcome, which is the Random Forest prediction.

4. KNN

The k-nearest neighbors algorithm, often known as KNN or k-NN, is a non-parametric, supervised machine learning classifier that employs vicinity to create classifications or predictions about an individual data point's grouping. Numerous progress has been made in the research on instance selection in the KNN study. However, although previously tests were performed on learning algorithms with the purpose of classification, instance selection on regression remains mostly unstudied [23]. Here in the KNN regression, the learner attempts to accurately identify the output for unseen cases after learning from the training set [23].

To illustrate the model's work, it begins with computing the distance between the test data and all of the training points, KNN attempts to predict the proper category for the test data. Then choose the K number of points that are closest to the test data. The probability of test data belonging to the classes of 'K' training data, and the class with the maximum probability is chosen, by the algorithm. The quantity in the case of regression is the mean of the 'K' chosen training points.

It begins by picking a location. If someone wanted to categorize the point, they had to determine the distance between the other points by using that point as a reference. Following that, it must locate the nearest neighbor by ordering them from shortest to greatest distance. Then it must vote based on the labels of the k closest neighbors.

The first step is to compute the distance between the new location and each training point. There are several ways for determining this distance, the most well-known of which are Euclidian, Manhattan (for continuous), and Hamming distances (for categorical). Here, Another aspect to note is the Minkowski distance. Minkowski distance is a metric in Normed vector space. A Normed vector space is a vector space on which a norm is defined.

The formula is below:

$$(\sum_{i=1}^{n}|x_i - y_i|^p)^{(1/p)}$$

Minkowski distance is the generalized distance metric. Here generalization means that we can manipulate the above formula to calculate the distance between two data points in different ways.

As aforementioned, we may alter the value of p and compute the distance in four ways: Manhattan Distance, p = 1
Euclidean Distance, p = 2
Hamming Distance, p = 3
Chebyshev Distance, p = ∞

To summarize, the k-nearest neighbor method seeks to determine the nearest neighbors of a given query location in order to provide a class label to that point. Although all training data is saved in memory, the algorithm adapts to account for any new data as additional training samples are introduced. When compared to other machine learning algorithms, KNN requires fewer hyperparameters because it simply requires a k value and a distance metric. However, it has its own drawbacks. KNN is a sloppy approach, as it consumes more memory and data storage than other classifiers and does not scale well. It suffers with high-dimensional data inputs. When the algorithm achieves the ideal number of features, the number of classification mistakes increases, especially when the sample size is small.

4.3.2 Deep Learning

Machine learning and deep learning are related topics in which machine learning and deep learning help artificial intelligence by giving a variety of skills and approaches and to resolve data-driven challenges, neural networks are used. Deep learning employs artificial neural networks that operate similarly to neural networks in the brain.

In view of the fact that the Neural Network primarily analyzes inputs in the forward direction, it is known as a Feed-Forward Network. A feed-forward network has no recollection of past activities; it always starts from the state it was previously taught to. Following the first training, there is no further progress. RNNs, on the contrary, get their own output from the previous application as well as all prior applications transitively. RNN is a Deep Learning algorithm

However, there is a concern known as long-term dependency.
Gradient Update Rule:

$$\text{New weight} = \text{weight - learning rate * gradient}$$

The vanishing gradient problem occurs when the gradient force as it back propagates over time of a gradient amount gets exceedingly low, which does not contribute much more to training, hence in recurrent no networks layers that receive a little gradient update do not benefit.

As a result, the LSTM solves the problem of long-term dependency. This network amplifies the RNN with a particular state known as an internal state. Although LSTM cells can train sequentially, they can capture long-term dependencies considerably more rapidly than typical RNNs. Various variants of LSTM have been created since its creation to improve on its current structures.

Long Short Term Memory(LSTM)

LSTM is an expression for long short-term memory networks, which are applied in Deep Learning. To get a better result in AQI prediction a long-short-term memory LSTM is merged into a recurrent neural network (RNN)-based model, to memorize previously viewed data [34].

Goodfellow [14] et.al. in their book said that a Recurrent neural network (RNN) has been found to be impressive in processing the sequential data, which exhibits some temporal sequence and whose value at each time-step depends on the context and requires remembering the context present in the data at the previous time-steps.

The fundamental ideas of LSTMs are cell states and their numerous gates. There are three gates: a forget gate, an input gate, and an output gate.

Now the forget gate specifies what kind of state information stored in this internal state here can be forgotten when it is no longer contextually relevant. The input gate then indicates what additional information we should add or update in the working storage state information. Then the output gate then specifies which parts of all the information contained in that state should be outputted in that particular instance.

Sigmoid activations σ_g are found in gates. It is comparable to the dRNN tanh in sigmoid activation σ_g, except instead of squishing values between +1 and -1, it squishes values between 0 and 1. This is useful for updating or forgetting data. Because any integer multiplied by 0 equals 0. Any integer multiplied by 1 has the same value, causing the values to vanish or be forgotten. As a result, that value remains constant or is preserved. This allows the network to learn what data should be deleted and what data should be kept.

The gates can be assigned numbers ranging from 0 to 1. Where a value of zero indicates that the gate is effectively closed and nothing can pass through, and a value of one indicates that the gate is wide open and everything may pass through it.

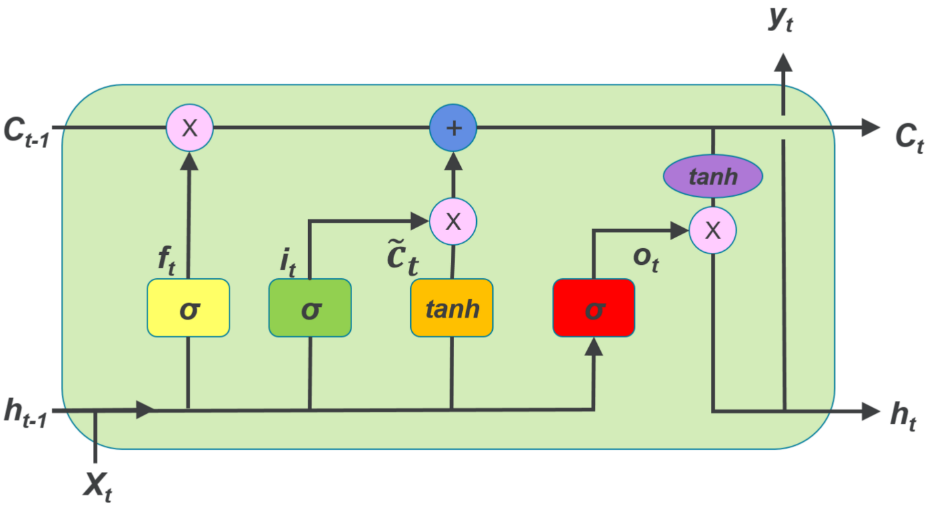


Figure 4.13: The LSTM model

It should be noted that the LSTM equations also create f_t, i_t, and $\bar{c}_t$ for internal use by the LSTM and are utilized to construct c_t and h_t.

$$f_t = \sigma_g(W_f \times x_t + U_f \times h_{t-1} + b_f)...(1)$$
$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{t-1} + b_i)...(2)$$
$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{t-1} + b_o)...(3)$$
$$\bar{c}_t = \sigma_c(W_c \times x_t + U_c \times h_{t-1} + b_c)...(4)$$
$$c_t = f_t.c_{t-1} + i_t \times .\bar{c}_t...(5)$$
$$h_t = o_t.\sigma_c(c_t)...(6)$$

Here,
σ_g : *sigmoid*
σ_c : tanh
. : element wise multiplication
f_t : is the forget gate
i_t : is the input gate

o_t : is the output gate
c_t : is the cell state(memory) at timestamp(t)
$\bar{c}_t$: represents candidate for cell state at timestamp(t)
h_t : is the hidden state

where ct signifies the LSTM cell state. The weights are $W_i, W_f, W_c, andW_o$, and the operator'.'represents the point wise multiplication of two vectors. When the cell state is updated, the input gate determines what new information may be stored in the cell state, and the output gate determines what information can be produced depending on the cell state [34].

The basis of LSTM is given by the equation in equation (1). The layer of the network, c_t, acts as LSTM memory and allows the RNN to maintain track of long-term dependencies.

As shown in Eq (6) the LSTM cell then decides to keep a fraction of its prior cell (h_{t-1}) state by using the forget gate f_t, illustrated in Eq (1). One more vital part, the input gate, determines how much new information is absorbed into the updated cell state from the prior cell output, as shown in i_t, Eq (2).
Ultimately, o_t represents the gate that holds the information necessary to update c_t, whereas h_t is the output of the LSTM unit at time t, as specified in Eq (5).
Despite classical RNN design, an LSTM cell calculates what more information to keep from former cells and how much to contribute to the ct or ongoing cell state. The fading gradient problem is considerably reduced by building a conduit for essential information to travel through and by connecting the gradient computations to the forget gate activations, f_t.

To avoid a project's sensitivity inflating or vanishing, LSTM or any version of LSTM may be effectively applied, which can forecast the reaction of the given input.

There are a few crucial items to take away from the following:
1. The calculations above are just for a single step. This signifies that the equations must be recalculated for the following time step. So, if we have a sequence of ten timesteps, the following equations will be calculated ten times for each timestep.
2. The biases (b_f, b_i, b_o, b_c) and weight matrices $(W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c)$ are not time-dependent. This means that the weight matrices remain constant from one time step to the next. In other words, the same weight matrices are employed to calculate the outputs of multiple timesteps [1].

4.3.3 Prediction Evaluation

The evaluation of a model's accuracy is a critical stage in every machine learning model. The Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-Squared (R^2), and Root Mean Squared Logarithmic Error (RMSLE) or Coefficient of prediction metrics are used to evaluate the model's performance in regression analysis.

1. RMSE

The square root of Mean Squared Error is Root Mean Squared Error (RMSE). Mean Squared Error (MSE) is the average of the squared difference between the data set's original and forecasted values. When a prediction is performed on a dataset, RMSE calculates the standard deviation of the residuals.

Hence,

$$RMSE = \sqrt{M}SE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

2. MAE

MAE or The Mean Absolute Error evaluates the average of the residuals in the dataset and indicates the average of the absolute difference between the actual and predicted values in the dataset.
It could be expressed as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

3. R^2

The coefficient of determination, often known as R-squared (R^2), shows the fraction of the variation in the dependent variable that the linear regression model explains. It is a scale-free score, which means that regardless matter whether the values are little or large, the value of R square will be less than one.
It is computed as follows:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

4. RMSLE

RMSLE or Root Mean Squared Logarithmic Error is the RMSE of the log-transformed predicted and target values. This measurement is beneficial when the target variable has a broad range and you do not want to compensate for large errors when the expected and target values are both high. It is also useful when you are concerned with percentage mistakes rather than exact values of errors.
The formula:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log y_i - \log \hat{y}_i)^2}$$

In the mentioned prediction metrics, y is the dependent variable. N is the number of data points, i is the model's prediction, $\hat{y}_i$ denotes the predicted value of y in the model, y_i denotes the target value, and $\bar{y}$ denotes the average or mean of observed data.

4.4 Software

These softwares are the most integral part for this project. Without the cohesion between each of them, it would not be possible for the flow to run thoroughly from start of the devices till the downloading of the datasets. All softwares were updated to their latest versions and ran on the supported hardware.

4.4.1 Arduino IDE

Arduino IDE is an open source software which is used in several platforms like windows, macOS, linux etc. to write codes for arduino and upload it directly to the board. The language used here is C. Several libraries have been developed and used to make all the sensors work in cohesion. Adafruit sensor library, DHT11 sensor library, DS3231 RTC library has been used to make the temperature and humidity sensor, date and time module, 3 MQ sensors pull data on to the serial monitor from the Arduino UNO R3 used in the project. It has several versions. The latest stable version which is 1.8.19 is used to ensure the most optimal outcome.

4.4.2 Raspberry Pi OS

As raspberry pi is a full fledged computer, it needs a proper operating system for a user to interact with it. Raspbian Buster is the newest version of the default OS for raspberry pi and it came out in 2019. It is designed for the best user experience along with the smoothest performance possible. File system access and browser compatibility have played a major role in this project, which was only possible for this operating system running in the background. It's freeware and is open-source software. The official website of Raspberry Pi carries the downloadable files for this OS.

4.4.3 Node RED

Node-RED is a flow-based programming tool that helps to describe an application's behavior. It is done via a network of black-boxes which are the main naming reason behind "Node"-RED. According to their official website, Raspbian Buster OS is the currently supported version of Raspberry Pi OS.

If a node is fed some data, it can be programmed to do something to pass that data on to another node, which can be an activity in a full system or just for visual representation. The programming and designing of the flows are done in a browser window.

4.4.4 Ngrok

Ngrok is a programmable network edge service that is used to establish secure and dependable connections between applications and systems. No port forwarding is essential which can be a hassle for people with no access to dedicated IP from their internet service provider. This project utilizes the simple codes of ngrok to connect users to the Raspberry Pi for the access of data via VNC Viewer.

Signing up with the service is the first step towards availing the services. Linux (ARM) version was selected as it is the one that is supported by the raspberry pi. An SSH tunnel is created then which can be accessed from anywhere.

4.4.5 VNC Viewer

VNC Viewer is an application which is used for the remote access of one supported device from another. It is supported on multiple platforms. Windows, MacOS, Raspbian Buster, Linux, Android, IOS are the notable ones. It can be used globally provided the network protocols are properly configured. The purpose of VNC viewer in this study is to give users the ability to download and access the dataset that is created and configured in the raspberry pi. VNC viewer requires an account set up and credentials provided by the owner of the account to the users. The address and port number is also necessary to establish a connection. The huge supported platform list is the main reason to pick this software for the access set up of the dataset from the host device.

4.4.6 Google Colab

Google colab is a web based IDE for python. It is for the benefit of running machine learning algorithms on the cloud monitored by multiple interest holders. The algorithms that are used for comparison in this study have been run through google colab and shared in between the research members.

Chapter 5

System Implementation

5.1 Device Connections and Build

The Arduino UNO, all the sensors, Raspberry Pi and how everything is composed together to form the data collection medium is discussed in this section.

5.1.1 Breadboard, Arduino and Sensors

The breadboard's positive and negative hole rails are powered by bringing in two male-to-male jumper wires from the arduino uno's 5V and GND pins and connecting them with one end of the strip respectively. It can be denoted as the power rail.

DHT11 Connection

To the breadboard's power rail, the temperature and humidity sensor DHT11's GND and VCC pins are connected and in Arduino uno R3, DAT pin is connected.

MQ-7 Connection

Carbon Monoxide sensor MQ-7's VCC and GND are connected to the breadboard's power rail and Analog out is connected to the A1 pin of the arduino.

MQ-131 Connection

Breadboard's power rail holds the VCC and GND pins of the MQ-131 and the Analog out has the connection to the arduino pin A3.

MQ-135 Connection

Power strip is connected to the VCC and GND pins of the NO_x sensor and Analog out is connected to the A2 pin of the arduino's analog pin rail.

DS3231 Connection

SCL and SDA connection of the DS3231 RTC module is connected to the PC4(SCL) and PC5(SDA) pins of the Arduino Uno. The power pins are connected to the breadboard. These pins need to be disconnected before uploading any sketch to the board.

PMS5003 Connection

The connection outlet of the PMS5003 sensor is connected to another extension PCB. Using Female to male jumper cables, the RXD pin has the connection to connected to the Arduino's TX(D0) pin. TXD has the connection to Arduino's RX(D1) pin. While uploading the sketch, this connection needs to be disconnected.

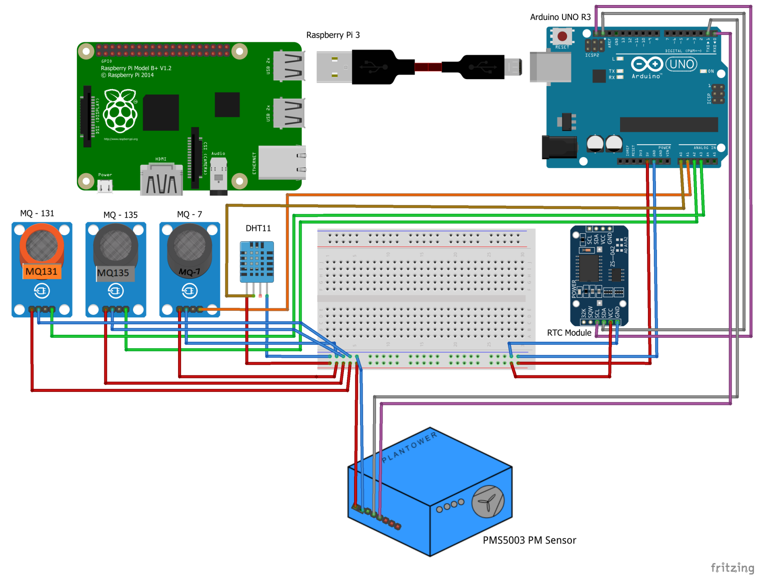


Figure 5.1: Sample of the proposed model

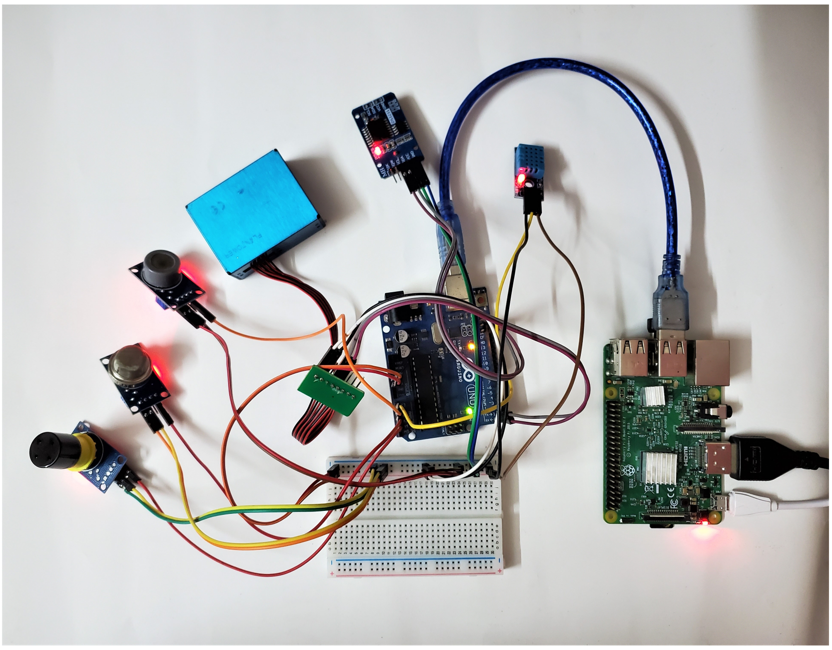


Figure 5.2: System model with interfacing all sensors

5.1.2 The Software Setup

1. Raspberry pi OS installation

The 64GB micro SD card was flashed with the latest Raspbian Buster OS from a different computer. The SD card is kept slotted in the card slot at the back of the Model B of Raspberry pi 3.

2. Node-RED, VNC viewer and ngrok installation

1. VNC Server and Viewer is installed with the OS package in the latest version of buster.
2. Node RED and ngrok is installed through the pi command line

3. Node-RED Setup

1. A serial Node is introduced from the list of nodes. It's an input node.
2. A function node, csv node and a msg.payload node is also created
3. The serial node is assigned to the port by which the Arduino Uno is connected to the Raspberry Pi.
4. The serial node brings in data from the arduino, separated by commas and sets up as the msg.payload node
5. The data transfer node returns the msg node which is connected to the the csv node in the flow.
6. The directory/home/pi/Downloads/spread.csv is addressed in the csv node which is why every data tuple which is spread.csv file.
7. To start the node RED, Applications menu Programming Node-RED - steps are followed. Also the command "node-red-start" can be used to turn it on. The command "node-red-off" can be run to stop the flow. After starting, the serial node will start to take data from the arduino's serial channel.

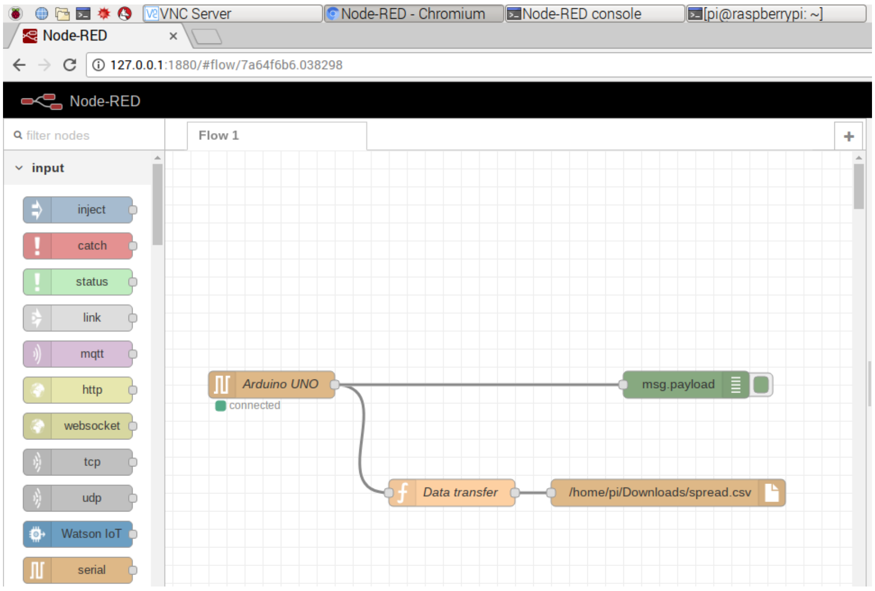
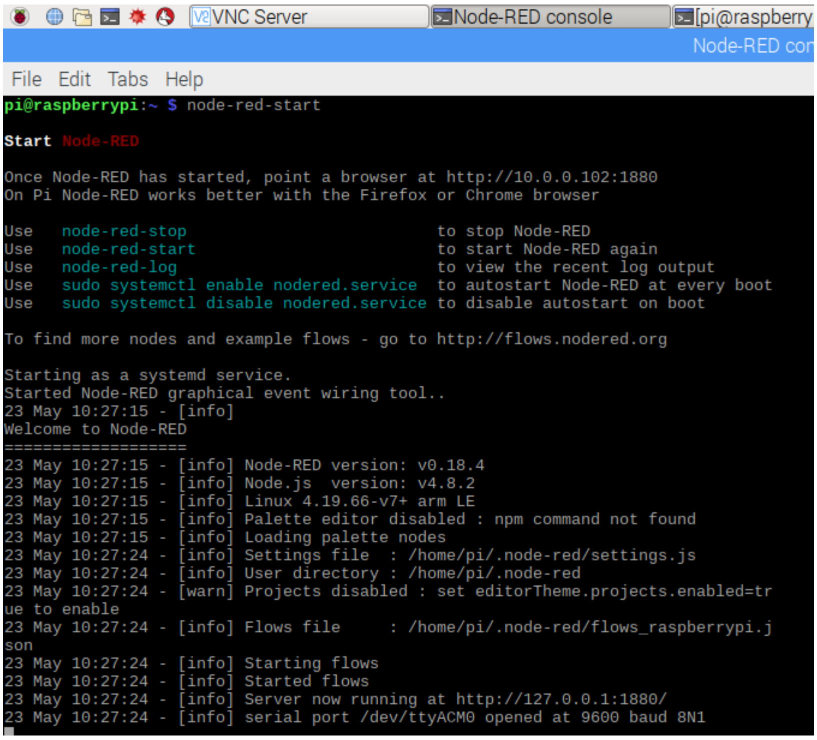


Figure 5.3: Node RED Flow



```
pi@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://10.0.0.102:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use   node-red-stop                          to stop Node-RED
Use   node-red-start                         to start Node-RED again
Use   node-red-log                           to view the recent log output
Use   sudo systemctl enable nodered.service  to autostart Node-RED at every boot
Use   sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
Started Node-RED graphical event wiring tool..
23 May 10:27:15 - [info]
Welcome to Node-RED
===================
23 May 10:27:15 - [info] Node-RED version: v0.18.4
23 May 10:27:15 - [info] Node.js  version: v4.8.2
23 May 10:27:15 - [info] Linux 4.19.66-v7+ arm LE
23 May 10:27:15 - [info] Palette editor disabled : npm command not found
23 May 10:27:15 - [info] Loading palette nodes
23 May 10:27:24 - [info] Settings file  : /home/pi/.node-red/settings.js
23 May 10:27:24 - [info] User directory : /home/pi/.node-red
23 May 10:27:24 - [warn] Projects disabled : set editorTheme.projects.enabled=tr
ue to enable
23 May 10:27:24 - [info] Flows file     : /home/pi/.node-red/flows_raspberrypi.j
son
23 May 10:27:24 - [info] Starting flows
23 May 10:27:24 - [info] Started flows
23 May 10:27:24 - [info] Server now running at http://127.0.0.1:1880/
23 May 10:27:24 - [info] serial port /dev/ttyACM0 opened at 9600 baud 8N1
```

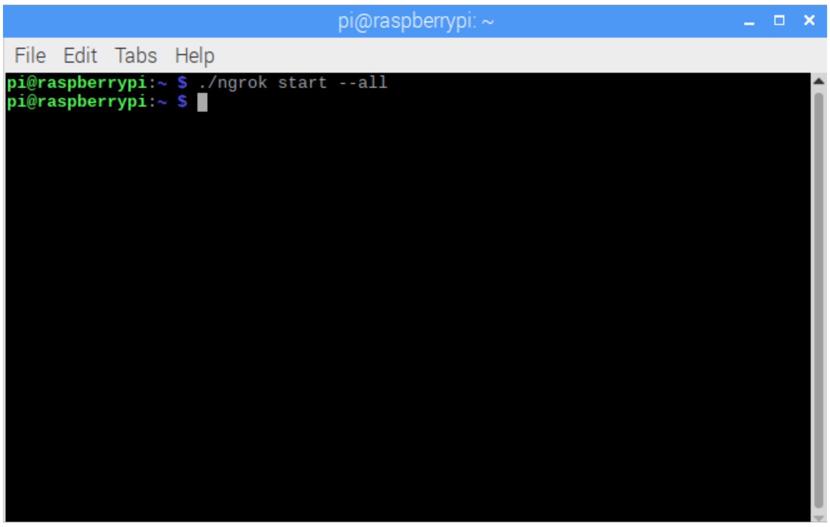
Figure 5.4: Node RED start command screen

4. Ngrok starting and setup

"./ngrok start –all" - command is typed into the command line of the raspberry pi

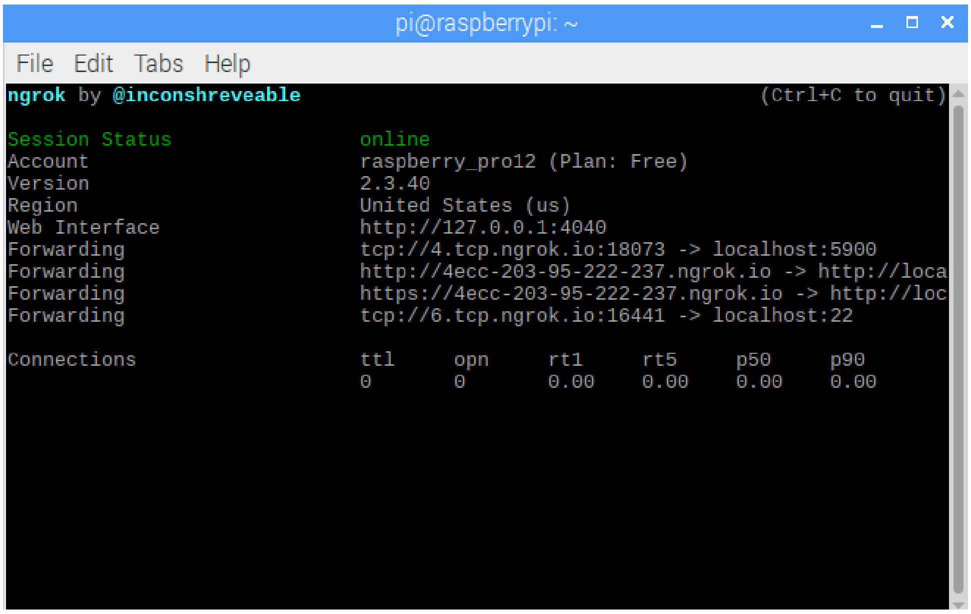
A session will be started under the free account created previously.

The forwarding address which ends in localhost:5900 is later to be inserted in the VNC viewer client module.



```
pi@raspberrypi:~ $ ./ngrok start --all
pi@raspberrypi:~ $ 
```

Figure 5.5: Ngrok starting command

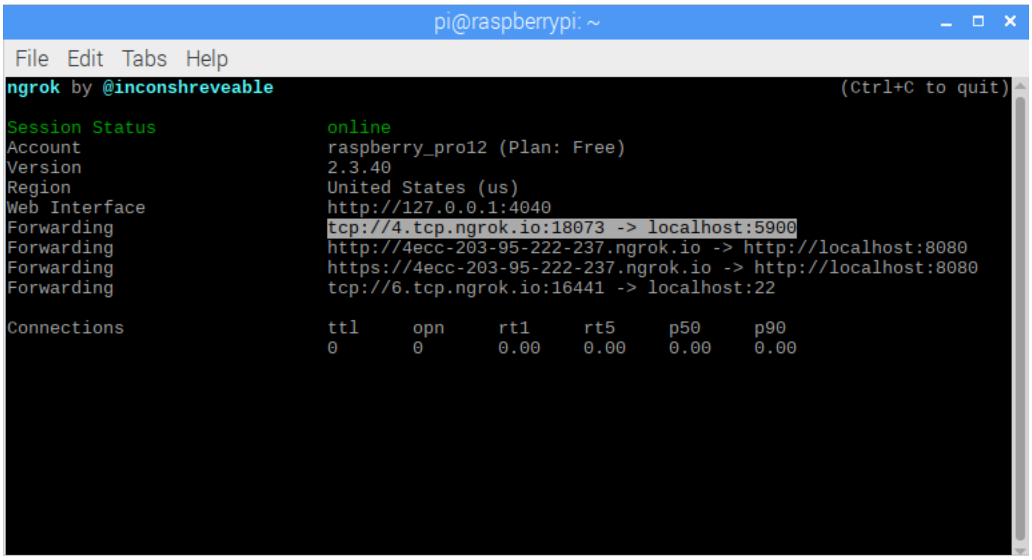


```
ngrok by @inconshreveable                                    (Ctrl+C to quit)

Session Status                online
Account                       raspberry_pro12 (Plan: Free)
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://4.tcp.ngrok.io:18073 -> localhost:5900
Forwarding                    http://4ecc-203-95-222-237.ngrok.io -> http://loca
Forwarding                    https://4ecc-203-95-222-237.ngrok.io -> http://loc
Forwarding                    tcp://6.tcp.ngrok.io:16441 -> localhost:22

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 5.6: Ngrok online command screen



```
ngrok by @inconshreveable                                                (Ctrl+C to quit)

Session Status                online
Account                       raspberry_pro12 (Plan: Free)
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://4.tcp.ngrok.io:18073 -> localhost:5900
Forwarding                    http://4ecc-203-95-222-237.ngrok.io -> http://localhost:8080
Forwarding                    https://4ecc-203-95-222-237.ngrok.io -> http://localhost:8080
Forwarding                    tcp://6.tcp.ngrok.io:16441 -> localhost:22

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 5.7: Highlighted Address to be copied by the client on their VNC viewer

5.1.3 Data Receiving Through Device

The step by step breakdown of the dataset procurement is described below:

1. Raspberry Pi

Turning the Raspberry Pi on via the AC adapter

Starting the Node-RED console from the Programme list

2. Arduino Uno

The arduino is connected to the Raspberry pi via an USB port

The serial channel's output is taken in via the USB Port

3. Node-RED

The outputs of the Debug window is the input of the Arduino Uno's Serial channel

This output is then processed and put into "spread.csv" file located in the - home/pi/downloads/ - directory

5.2 Dataset

The creation, maintenance and access of the required dataset is the primary and sole goal of this research and hence this research worked with two different datasets. Let the compared dataset be Dataset A and the proposed dataset be Dataset B.

5.2.1 Source

As this research worked with two different datasets so the sources of these datasets are different. The datasets are elaborated below.

Dataset A

Dataset A is the dataset that was created manually from the existing information. The monthly average information of P$M_{2.5}$, O_3, NO_2, CO was taken from the following website: http://case.doe.gov.bd/. On the other hand, the information of daily Humidity and Temperature was taken from https://en.tutiempo.net/ this website. The monthly average data of P$M_{2.5}$, O_3, NO_2, CO were taken and so, the monthly average values of Humidity and Temperature was also calculated from the given daily values.

Dataset B

The implemented IoT device in this comprehensive research, is the source and it is the main procuring medium of the acquired dataset.

5.2.2 Dataset Description

Dataset A

The website from where the data of P$M_{2.5}$, O_3, NO_2, CO were procured were from the month of November 2012 to July 2018. After that no recent data was provided by this website and so the Humidity and Temperature were also taken also from November 2012 to July 2018. After taking the values manually the file was then converted into a csv file.

The tabular date and time information of the dataset is given below:

Total number of Years	5.9
Total number of Months	70

Table 5.1: Summary of the Dataset's distinction (Dataset A)

Dataset B

The procured data from the IoT which was proposed for this research was taken from 2 May, 2022 to 15 May, 2022. The IoT device took the hourly data for these 14 days.

The time date information was divided into 6 different columns. The six different columns were accordingly: Year, Month, Day, Hour, Minute, Second. The file was already in a csv format as it was downloaded from the IoT device.

The tabular date and time information of the dataset is given below:

Total number of Days	14
Total number of Hours	336
Hour interval	1

Table 5.2: Summary of the Dataset's distinction (Dataset B)

Additionally, there are six columns in the dataset which consist of the collected air components which are : NO_x, CO, O_3, P$M_{2.5}$, P$M_{1.0}$, PM_{10}. For this research we also took two columns for the meteorological components: Humidity and Temperature.

For each day, each of the components took the data from the 0:00 to 23:00 in a total of 24hours. So, the csv file incorporates the data of a total 336 hours.

5.2.3 Data Sample

Dataset A

	Date (yyyy/m/d)	PM2.5/ AQI (µg /m3)	NO2 (ppb)	CO (ppm)	O3 (ppb)	Temperature (C)	Humidity (%)
1	2012/11/30	109.11	25.12	1.41	23.03	20.9	71
2	2012/12/31	188.69	26.88	2.37	7.63	17.8	
3	2013/01/31	227.42	39.8	3.54	9.17	17.29	68.01
4	2013/02/28	127.7	16.01	0.28	8.12	24.8	46
5	2013/03/31	88.06	7.99	0.44	5.31	29.4	66
.	....		....	...	....	....	..
.	....		....	...	....	....	..
68	2018/5/31	108	17	2.67	2.33	26.1	88
69	2018/6/30	93	5.34	2.81	1.78		
70	2018/7/31	105	6.85	3.45	1.85	29.7	78

Table 5.3: Sample Data of Data set A

Here, the monthly average data of the air components from 30-11-2012 to 31-07-2018 can be seen.

Dataset B

	Year	Month	Day	Hour	Minute	Second	Humidity	Temperature	CO	NOx	O3	PM 1.0	PM 2.5	PM 10	AQI
1	2022	5	2	0	51	6	78	36	287	397	178	138	145	142	145
2	2022	5	2	1	51	6	88	28	291	391	176	139	151	152	151
3	2022	5	2	2	51	11	77	37	294	404	176	136	148	150	148
4	2022	5	2	3	51	16	74	29	290	399	174	136	148	150	148
5	2022	5	2	4	51	21	86	32	293	382	177	136	148	150	148
.	....	.	.	.	..	..	..	..	...	...	...	...	...	...	...
.	....	.	.	.	..	..	..	..	...	...	...	...	...	...	...
334	2022	5	15	21	11	59	77	37	238	343	154	131	151	161	151
335	2022	5	15	22	12	4	72	31	238	342	151	131	151	161	151
336	2022	5	15	23	31	37	82	34	237	337	149	136	159	170	159

Table 5.4: Sample Data of Dataset B

Here, the data from 2-05-2022 to 15-05-2022 of the air components is shown. For every hour, the readings from the sensors were saved to their corresponding columns.

5.3 Data Pre-processing

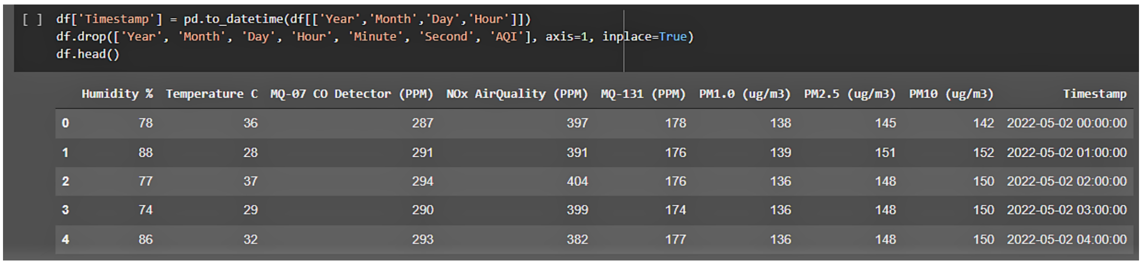
The proposed system is taking data tuples in such a way that for running the dataset through most machine learning algorithms and deep learning algorithms, the preprocessing would already be done for it. If not completely done, most of it would be very basic manipulations and conjunctions. The procured dataset is pre-processed following two slightly different ways for the algorithms chosen for this study.

5.3.1 Data Pre-processing for Deep Learning Model

Dataset B

Hossain [38] et. al. in their research mentioned that as LSTM is one of the layers of deep learning that can make a robust model to be used for predicting AQI. The procured dataset has been tailored from the very beginning to fit in its requirements perfectly. The year, month, day, hour, minute, second etc. is readily available without any splitting of data entries.

With every set intervaled iteration of the arduino's data intake, the Node-RED code is set to create a data entry with all the date and time individually as different entries. Hence, with the Timestamp the dataset is then indexed for further model LSTM implementation.



```
[ ] df['Timestamp'] = pd.to_datetime(df[['Year','Month','Day','Hour']])
    df.drop(['Year', 'Month', 'Day', 'Hour', 'Minute', 'Second', 'AQI'], axis=1, inplace=True)
    df.head()
```

	Humidity %	Temperature C	MQ-07 CO Detector (PPM)	NOx AirQuality (PPM)	MQ-131 (PPM)	PM1.0 (ug/m3)	PM2.5 (ug/m3)	PM10 (ug/m3)	Timestamp
0	78	36	287	397	178	138	145	142	2022-05-02 00:00:00
1	88	28	291	391	176	139	151	152	2022-05-02 01:00:00
2	77	37	294	404	176	136	148	150	2022-05-02 02:00:00
3	74	29	290	399	174	136	148	150	2022-05-02 03:00:00
4	86	32	293	382	177	136	148	150	2022-05-02 04:00:00

Figure 5.8: Dataset with Timestamp indexing

Now, after plotting the components in the graphs, the graphs showed that index 48 had the wrong timestamp.

Here,
X axis = Date
Y axis = The values of the corresponding components

The graph is shown below:

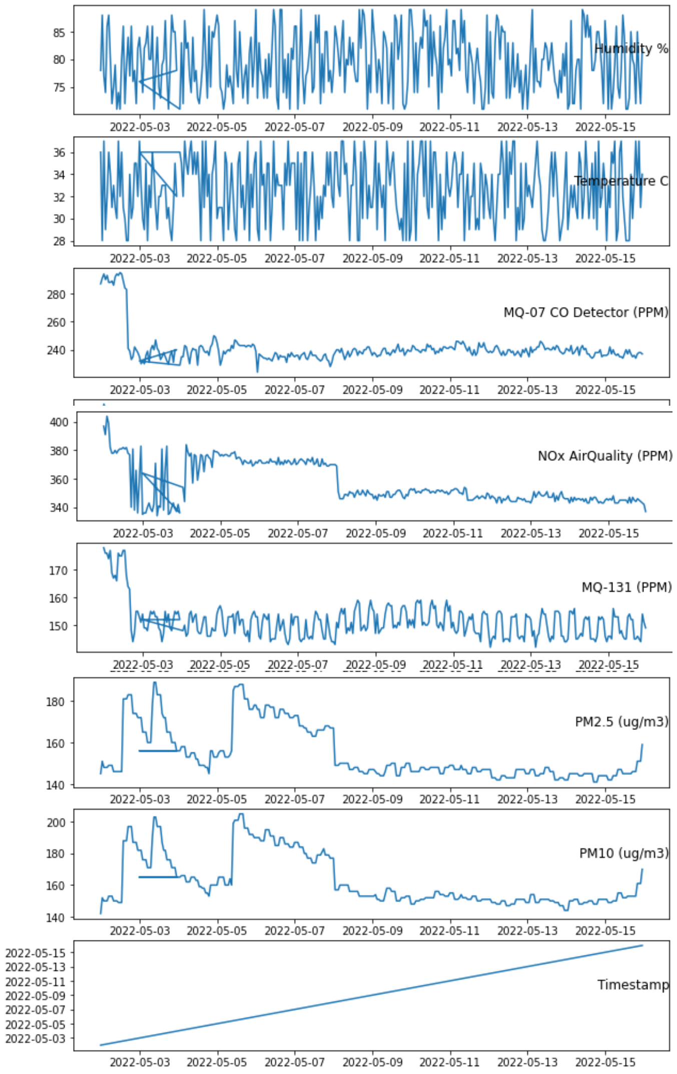
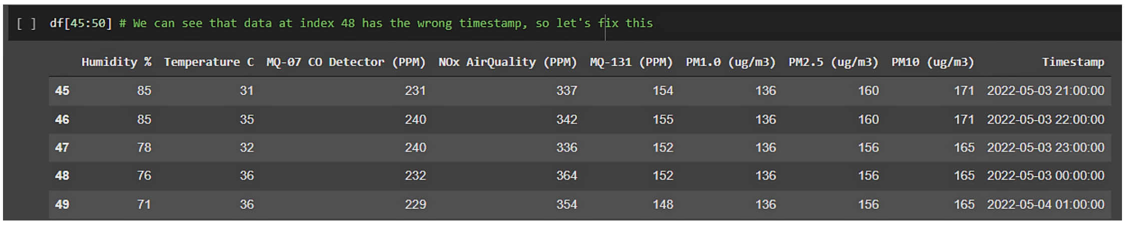


Figure 5.9: The graphs of Air Components

More precisely,

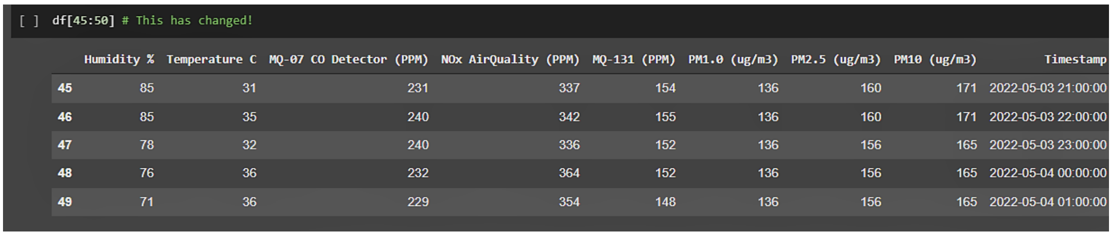


```
[ ] df[45:50] # We can see that data at index 48 has the wrong timestamp, so let's fix this
```

	Humidity %	Temperature C	MQ-07 CO Detector (PPM)	NOx AirQuality (PPM)	MQ-131 (PPM)	PM1.0 (ug/m3)	PM2.5 (ug/m3)	PM10 (ug/m3)	Timestamp
45	85	31	231	337	154	136	160	171	2022-05-03 21:00:00
46	85	35	240	342	155	136	160	171	2022-05-03 22:00:00
47	78	32	240	336	152	136	156	165	2022-05-03 23:00:00
48	76	36	232	364	152	136	156	165	2022-05-03 00:00:00
49	71	36	229	354	148	136	156	165	2022-05-04 01:00:00

Figure 5.10: Timestamp error at index 48

After fixing the error that occurred at index 48 the fixed dataset and the fixed graphs are shown below:



```
[ ] df[45:50] # This has changed!
```

	Humidity %	Temperature C	MQ-07 CO Detector (PPM)	NOx AirQuality (PPM)	MQ-131 (PPM)	PM1.0 (ug/m3)	PM2.5 (ug/m3)	PM10 (ug/m3)	Timestamp
45	85	31	231	337	154	136	160	171	2022-05-03 21:00:00
46	85	35	240	342	155	136	160	171	2022-05-03 22:00:00
47	78	32	240	336	152	136	156	165	2022-05-03 23:00:00
48	76	36	232	364	152	136	156	165	2022-05-04 00:00:00
49	71	36	229	354	148	136	156	165	2022-05-04 01:00:00

Figure 5.11: Timestamp fixed Timestamp at index 48

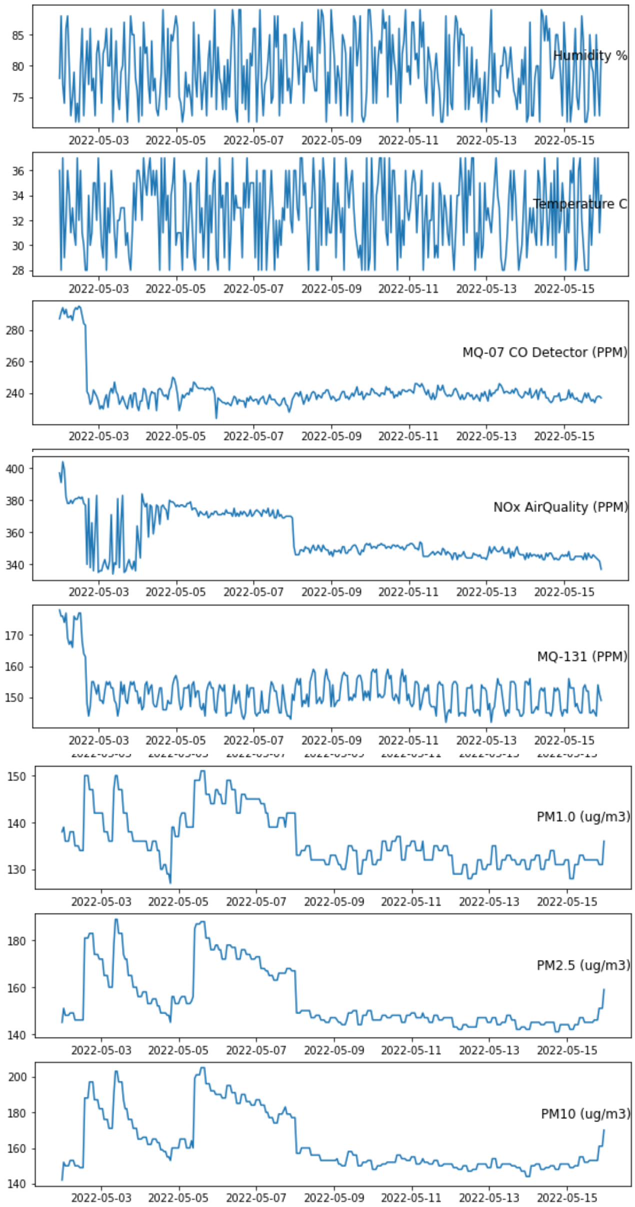


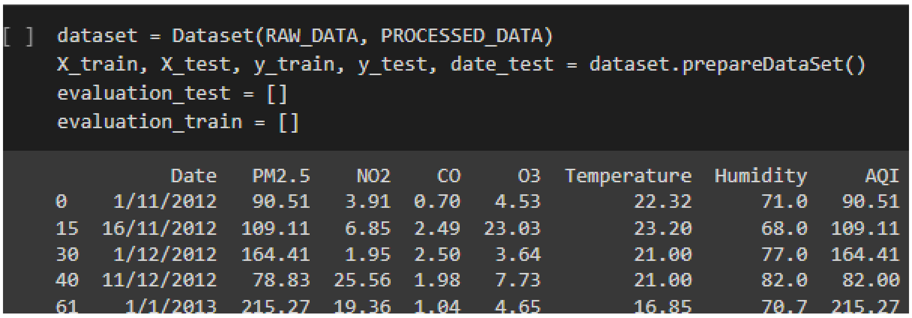
Figure 5.12: The fixed graph of Air Components

Thus by pre-processing the error that was made by the timestamp was fixed the dataset became more robust for the further analysis.

5.3.2 Data Pre-processing for Regression Model

Dataset A

In terms of the pre-processing of regression models, at first the csv file of Dataset A was converted to a pre-processed csv file where the AQI is calculated and added in a column. The sample of the pre-processed dataset is given below:



```
[ ]  dataset = Dataset(RAW_DATA, PROCESSED_DATA)
     X_train, X_test, y_train, y_test, date_test = dataset.prepareDataSet()
     evaluation_test = []
     evaluation_train = []
```

```
              Date   PM2.5    NO2    CO     O3  Temperature  Humidity     AQI
0      1/11/2012   90.51   3.91  0.70   4.53        22.32      71.0   90.51
15    16/11/2012  109.11   6.85  2.49  23.03        23.20      68.0  109.11
30     1/12/2012  164.41   1.95  2.50   3.64        21.00      77.0  164.41
40    11/12/2012   78.83  25.56  1.98   7.73        21.00      82.0   82.00
61      1/1/2013  215.27  19.36  1.04   4.65        16.85      70.7  215.27
```

Figure 5.13: Pre-processed Dataset A

Moreover, as the new pre-processed dataset had a new column added "AQI" hence, AQI is plotted in a graph.

Here,
X axis = Date
Y axis = The values of AQI

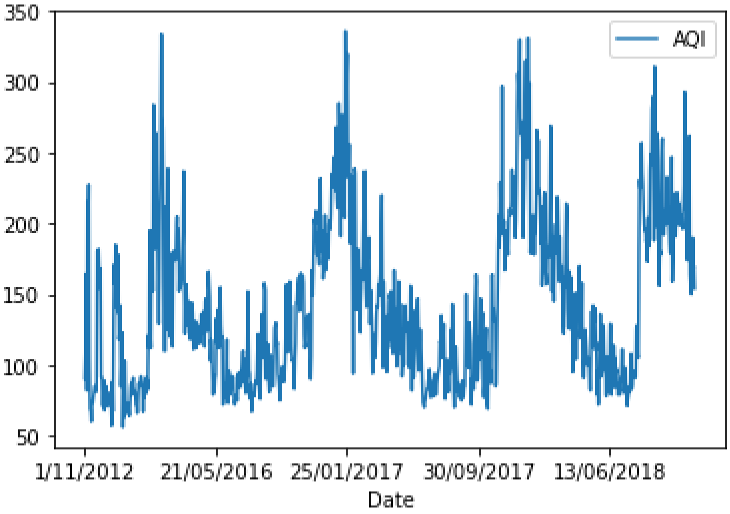


Figure 5.14: The graph of AQI for Dataset A

The scattered plot of dataset A is given below:

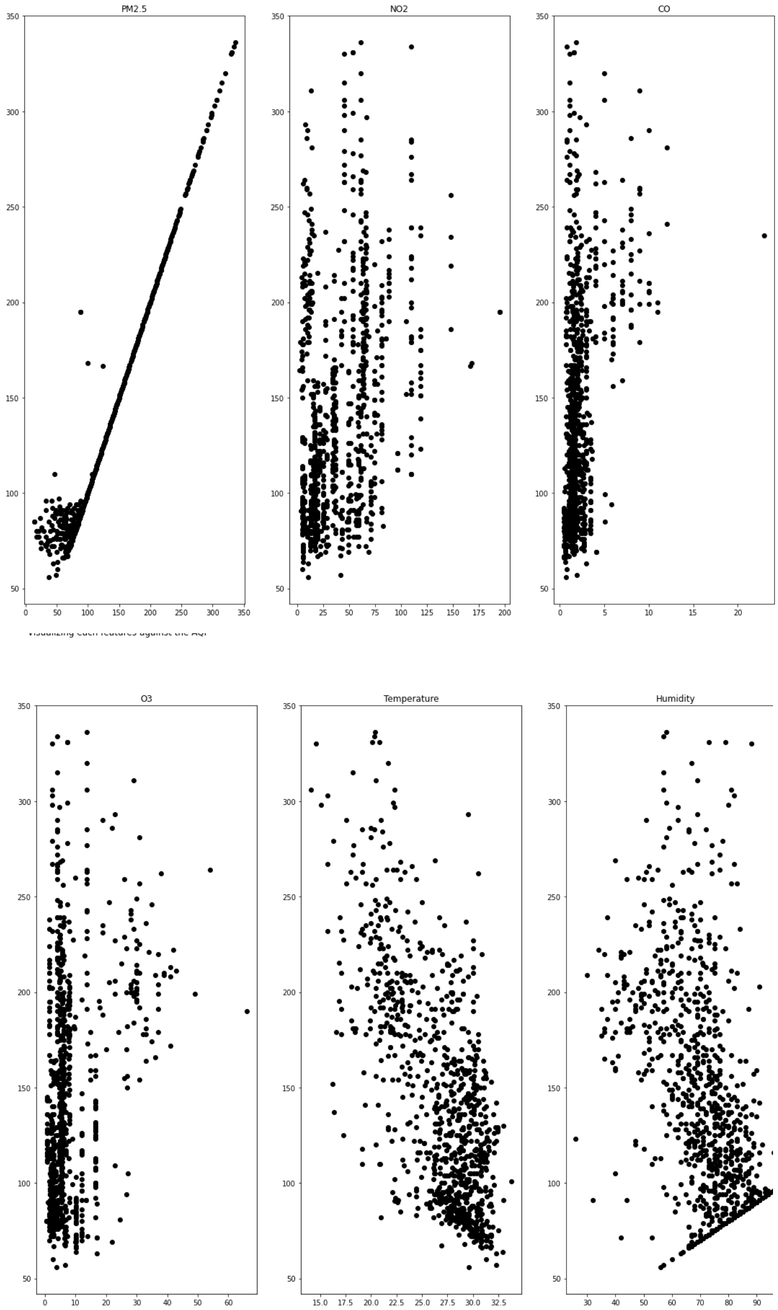


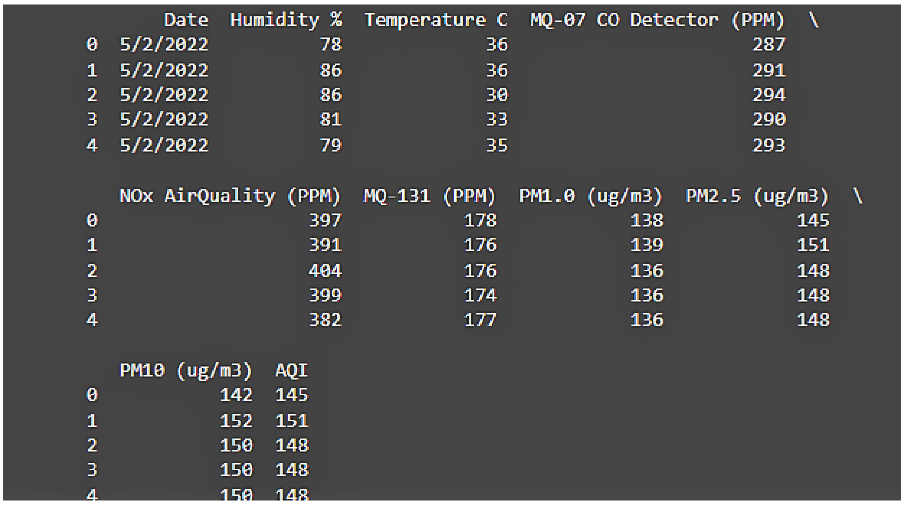
Figure 5.15: The scattered plot of Air Components for Dataset A

The scattered plot explains that the correlation for each component against the AQI is different. Different plots for different components have been shown but if it is noticed at P$M_{2.5}$, the scattered plot is the least scattered against AQI. So, this scattered plot indicates that there is some sort of correlation between AQI and P$M_{2.5}$.

Dataset B

In terms of the pre-processing of regression models, at first the raw data from the IoT device was converted to a pre-processed csv file where the AQI is calculated and added in a column. Secondly, the proposed dataset which was procured from the device had the format of Date and Time in individual 6 columns: Year, Month, Day, Hour, Minute, Second which we mentioned before. As for the amelioration of the model this format of the timestamp was therefore converted into a single column named "Date" consisting of Day, Month and Year.

The sample of the pre-processed dataset is given below:



```
        Date  Humidity %  Temperature C  MQ-07 CO Detector (PPM)  \
0  5/2/2022          78             36                      287
1  5/2/2022          86             36                      291
2  5/2/2022          86             30                      294
3  5/2/2022          81             33                      290
4  5/2/2022          79             35                      293

   NOx AirQuality (PPM)  MQ-131 (PPM)  PM1.0 (ug/m3)  PM2.5 (ug/m3)  \
0                   397           178            138            145
1                   391           176            139            151
2                   404           176            136            148
3                   399           174            136            148
4                   382           177            136            148

   PM10 (ug/m3)  AQI
0           142  145
1           152  151
2           150  148
3           150  148
4           150  148
```

Figure 5.16: The Pre-processed Dataset B

Moreover, as the new pre-processed dataset had a new column added "AQI" hence, AQI is plotted in a graph.
Here,
X axis = Date
Y axis = The values of AQI

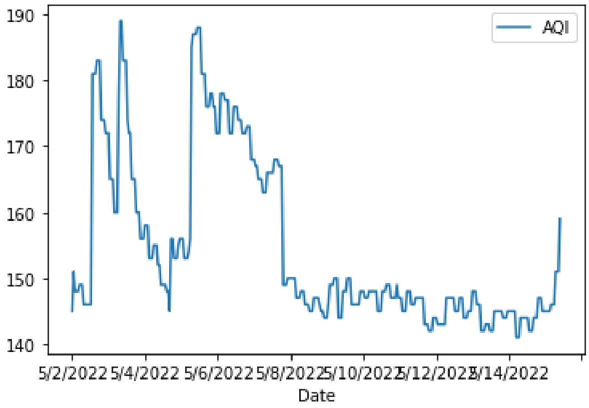


Figure 5.17: The Graph of AQI for Dataset B

The scatter plot of the components against AQI is also showed:

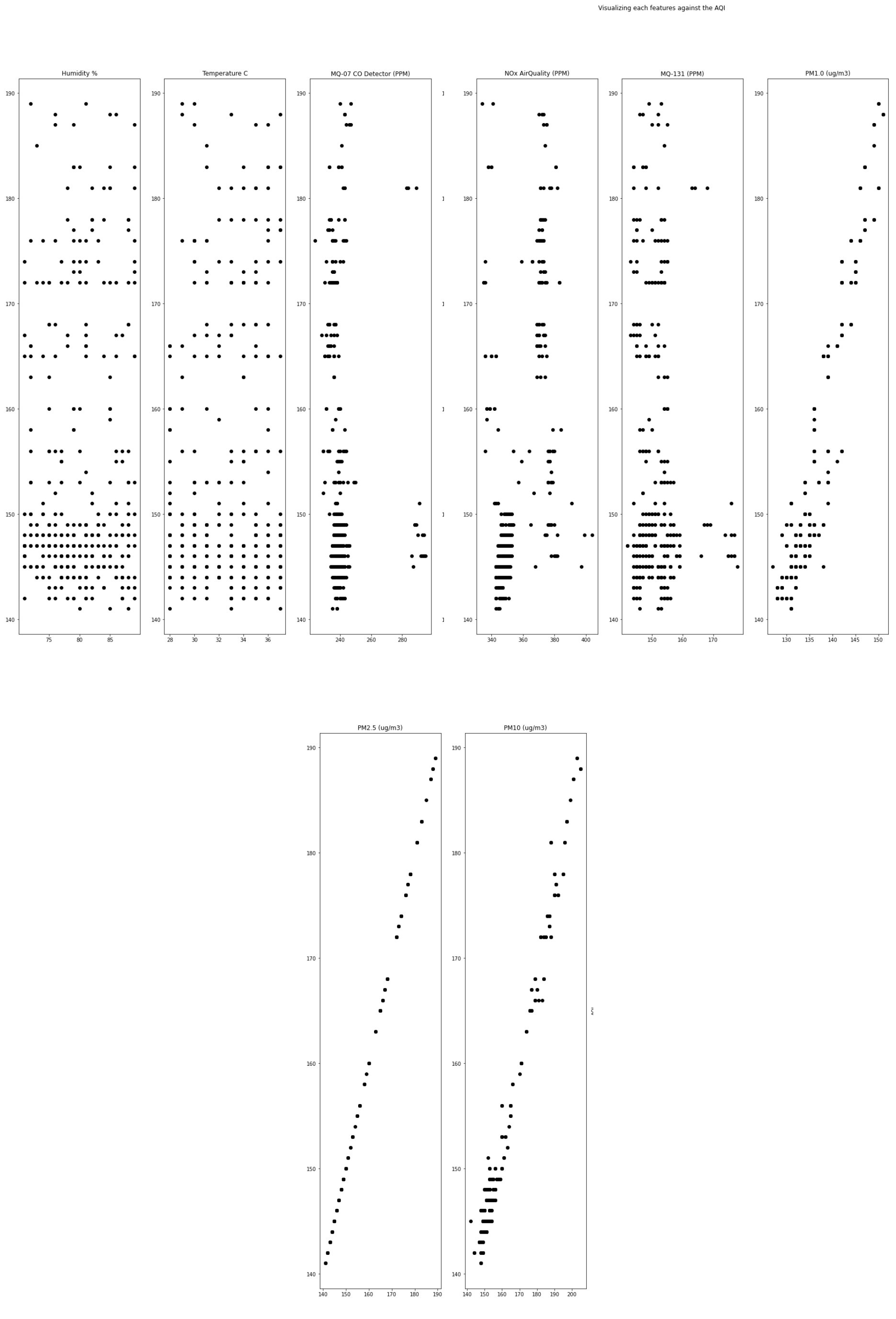


Figure 5.18: The scattered plot of Air Components for Dataset B

The scattered plot explains that the correlation for each component against the AQI is different. Different scattered plots for different components are shown here but if P$M_{2.5}$ is noticed, the scattered plot is the least scattered against AQI. So, this scattered plot indicates that there is some sort of correlation between AQI and P$M_{2.5}$.

As demonstrated, in this way the transformation that preprocessing did to the raw dataset, made it more efficient.

5.4 Time Series Forecasting Models

5.4.1 Regression Models

Linear Regression, Random Forest, KNN and Lasso Regression were the models for both Dataset A and B. The data has been split into 80:20 ratio for training and testing all the models. Then in order to evaluate the prediction of the models several loss metrics have been used to make a comparison between the predictions of the models.

5.4.2 Deep Learning Models

For more precise control over the model single layer LSTM model and 2 layers LSTM model have been used. The data has been split into 90:10 for the train and test both models.

Here, this particular study predicted an hour to the future by taking sampling rate = 1. To predict the next hour the model needed to look back at the last 5 hours by taking the length = 5. In this research, the prediction was done for all the features. Now, for one iteration the number of samples was taken = 16.

5.4.3 Single Layer LSTM

This single layer LSTM model has a unit of 4. In deep learning models, the networks need to comprehend the relationship properly between the input and output layers, hence with the help of the hyperparameter “Activation” function, the networks can learn properly. In this study “Relu” was used as the activation. A “Dense” layer with 8 output neurons to change the dimensionality of the output was added. The number of epochs was fixed to 100. Then depending on the prediction a loss metric RMSE provided its scores for the train and test.

5.4.4 Double Layer LSTM

This stacked double layer LSTM model has a unit of 64 for the first layer and unit of 32 for the second layer. In this model the activation function which was used was “Relu” for the first and second layers. Additionally, a “Dense” layer with 8 output neurons to change the dimensionality of the output was added. Another layer, “Dropout” was added which is known as a regularization technique and is used to prevent overfitting. For this study Dropout = 0.2 was taken. The number of epochs was fixed to 100. Then depending on the prediction a loss metric RMSE provided its scores for the train and test.

Inputs	
Sampling Rate	1
Length/Look Back	5
Batch Size	16
Single Layer LSTM	
Unit of LSTM	4
Activation Function	Relu
Epoch	100
Dense Layer Output	8
Double Layer LSTM	
First Layer Unit	64
Second Layer Unit	32
Activation Function	Relu
Epoch	100
Dropout	0.2
Dense Layer Output	8

Table 5.5: Hyperparameters setting of LSTM models.

Chapter 6

Results and Analysis

6.1 Accessing Dataset from Remote Device

Through the implementation of VNC server and viewer, users/clients having the application VNC viewer on their respective devices can download the CSV file with ease. They just need to have the connecting address of the TCP protocol from the ngrok command line in the arduino uno.

6.1.1 VNC Viewer setup and access

For cloud access for the csv file, VNC Server is set up in the Raspberry Pi.

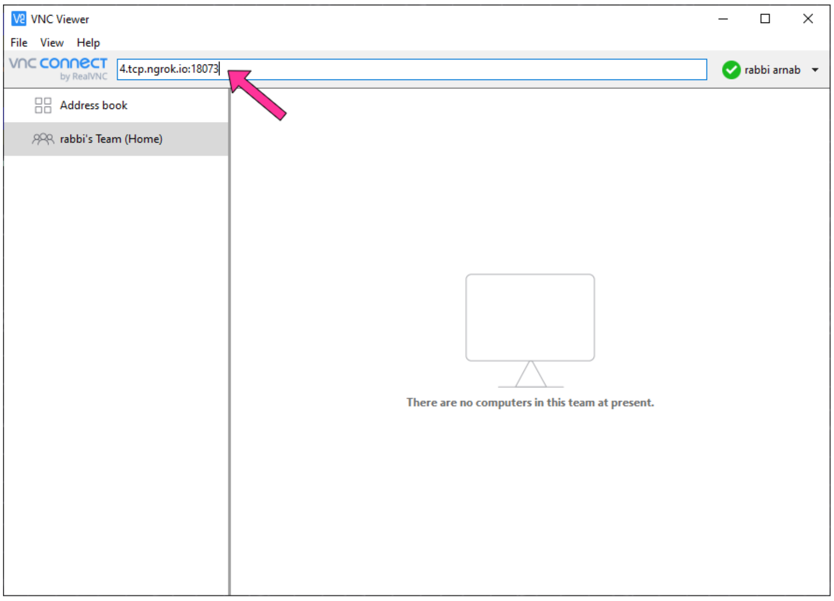
Client/User puts in the address shown in the ngrok command screen which ends in localhost:5900.

It requires a set of credentials to log into the server. The default username and password is set at “pi” and “raspberry” respectively.

File transfer is selected from the remote desktop inside the raspberry pi remote window.

The directory /home/pi/Downloads/ is navigated to by the user and the download option is selected. The desired directory to be downloaded to is chosen on the left side of the VNC window.

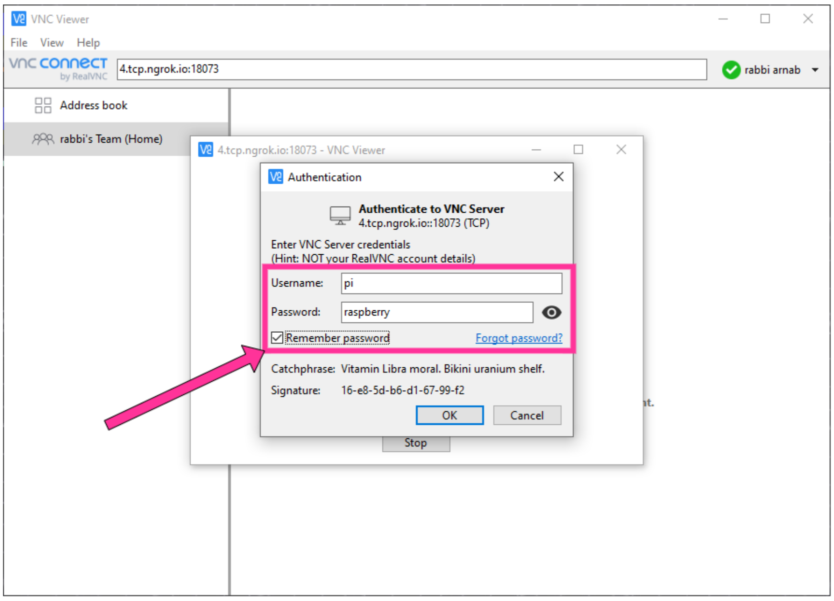
The CSV file is then downloaded over the cloud in the user’s device.



4.tcp.ngrok.io:18073

There are no computers in this team at present.

Figure 6.1: VNC Viewer client connection screen.



4.tcp.ngrok.io:18073

Authenticate to VNC Server
4.tcp.ngrok.io::18073 (TCP)

Enter VNC Server credentials
(Hint: NOT your RealVNC account details)

Username: pi

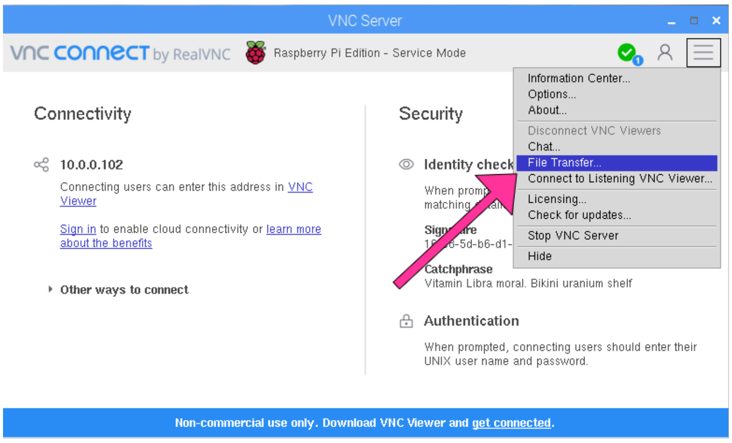
Password: raspberry

☑ Remember password   Forgot password?

Catchphrase: Vitamin Libra moral. Bikini uranium shelf.

Signature: 16-e8-5d-b6-d1-67-99-f2

Figure 6.2: VNC Viewer Client Credential Entry.



Connectivity

10.0.0.102

Connecting users can enter this address in VNC Viewer

Sign in to enable cloud connectivity or learn more about the benefits

Other ways to connect

Security

Identity check

When promp… matching …

Signature
1…6-5d-b6-d1-…

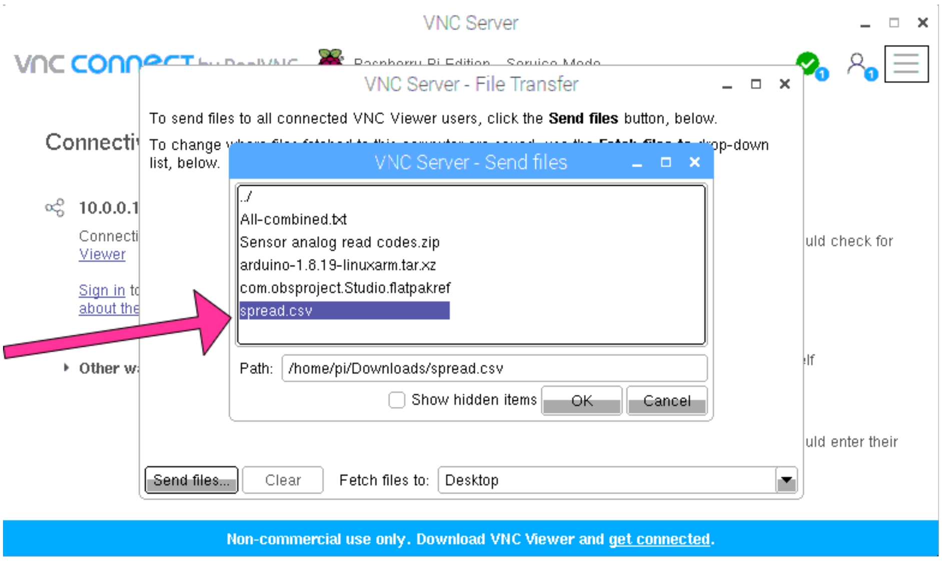
Catchphrase
Vitamin Libra moral. Bikini uranium shelf

Authentication

When prompted, connecting users should enter their UNIX user name and password.

Non-commercial use only. Download VNC Viewer and get connected.

Figure 6.3: File Transfer initiation after the user has access to the remote server.



To send files to all connected VNC Viewer users, click the **Send files** button, below.

To change where files fetched to this computer are saved, use the **Fetch files to** drop-down list, below.

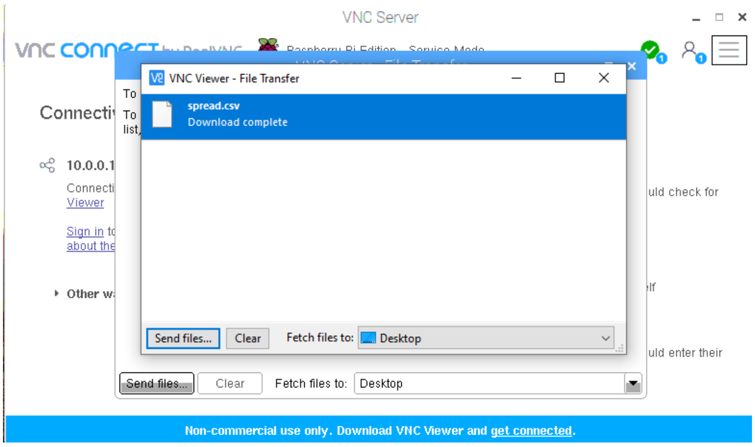
```
../
All-combined.txt
Sensor analog read codes.zip
arduino-1.8.19-linuxarm.tar.xz
com.obsproject.Studio.flatpakref
spread.csv
```

Path: /home/pi/Downloads/spread.csv

Fetch files to: Desktop

Non-commercial use only. Download VNC Viewer and get connected.

Figure 6.4: Selection of the CSV to be downloaded.



spread.csv
Download complete

Fetch files to: Desktop

Non-commercial use only. Download VNC Viewer and get connected.

Figure 6.5: Client side download complete notification.

6.2 Time Series Forecasting Models

The results of the forecasting models on AQI components are evaluated based on the given scores of the loss metrics.

The results of Dataset A and Dataset B by the prediction models are shown below:

6.2.1 Regression Models Results

Train Evaluation For Dataset A

Regression Models	RMSE	MAE	R^2	RMSLE
Linear	95.079796023	6.1843945882	0.97047731350	0.0149010630
Random Forest	3.8638305977	0.2675120967	0.99880026394	0.0001774989
KNN	11.381106033	2.0206541218	0.99646611751	0.0005740135
Lasso	97.947781615	6.4937172680	0.96958679161	0.0144993803

Train Evaluation For Dataset B

Regression Models	RMSE	MAE	R^2	RMSLE
Linear	1.3895255013	3.1921397531	1.0	2.0310224e-31
Random Forest	0.0072175373	0.0354104477	0.99995749283	2.3866692e-07
KNN	1.0737976782	0.7686567164	0.99367594594	4.2945891e-29
Lasso	0.9999999999	0.8498743817	0.99411057207	3.6071520e-05

Table 6.1: Train Evaluation

According to the train evaluation of Dataset A, which was created manually with the existing data, it shows that the RMSE, RMSLE, MAE error metrics which work to evaluate the error level in terms of Random Forest are the lowest. The lower the values of these metrics, the higher the accuracy of time forecasting. On the other hand, the Lasso Regression model shows the highest value of RMSE and MAE and the second-highest value of R^2. Then if the other two models are noticed, it can be seen that the loss metrics RMSE, RMSLE, and MAE for KNN are better than the Linear Regression model. Then again, the R^2 metric for the Linear Regression model is less than the KNN model.

On the contrary, according to the train evaluation of Dataset B, which is the proposed dataset of this research, it shows that the RMSE, and MAE error metrics which work to evaluate the error level in terms of Random Forest are the lowest but the RMSLE and R^2 are greater than the other models. On the other hand, Linear has the lowest RMSLE but the MAE, R^2 and RMSE values are higher than the other models.Now to compare the proposed dataset with the manual dataset, it can be seen that the Random Forest Regression shows better results on average than the other models.

Test Evaluation For Dataset A

Regression Models	**RMSE**	**MAE**	R^2	**RMSLE**
Linear	137.76886307	6.7867847706	0.96165680889	0.01789861023
Random Forest	21.633446313	0.8533582887	0.99397907954	0.00085596078
KNN	20.529208199	2.8323707664	0.99428640597	0.00080302636
Lasso	144.36470314	7.09878065321	0.95982108526	0.01737644623

Test Evaluation For Dataset B

Regression Models	**RMSE**	**MAE**	R^2	**RMSLE**
Linear	1.1285351302	2.7167810e-14	1.0	1.5081164e-31
Random Forest	0.0127441176	0.0538235294	0.99991073598	4.7817862e-07
KNN	1.3986928104	0.9215686274	0.99020309293	5.7314589e-29
Lasso	0.8549888221	0.7847251738	0.99401137550	3.1422085e-05

Table 6.2: Test Evaluation

As per the test evaluation of Dataset A, which was constructed manually using existing data, the outcome is worse than the train evaluation. In train evaluation the value of RMSE for Linear Regression model is 95.07 but the test evaluation of RMSE for Linear Regression model is 137.77. If the other models are noticed, it can be seen that all the loss metrics values are higher in the test evaluation result than the train evaluation result. The most possible reason for this worse test evaluation result is as the data were taken from multiple websites and evaluated based on the monthly average data so there is a severe fluctuation in the dataset.

On a different note, the test evaluation of Dataset B shows lower loss metrics values for Linear Regression and Lasso Regression models than their train evaluation values. The other test evaluation results also do not show much difference as the proposed dataset retrieved from the device was better than the Dataset A.

The histograms of the Regression models are shown below where,

Dataset A

X axis = frequency
Y axis = AQI

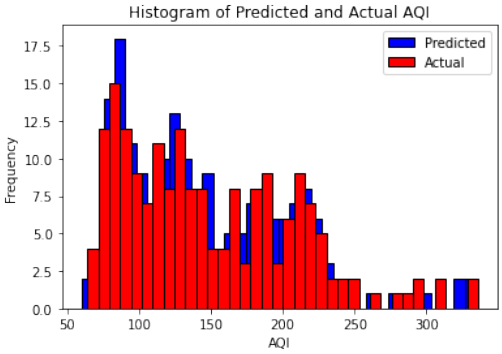


Figure 6.6: The Histogram of Linear Regression For Dataset A

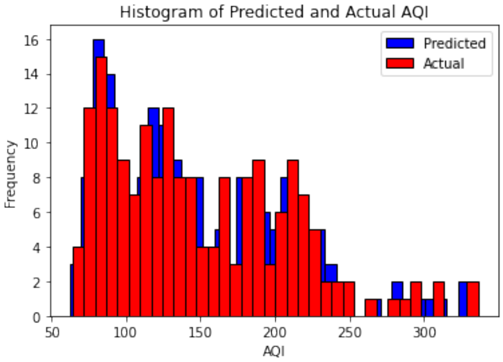


Figure 6.7: The Histogram of Random Forest Regression For Dataset A

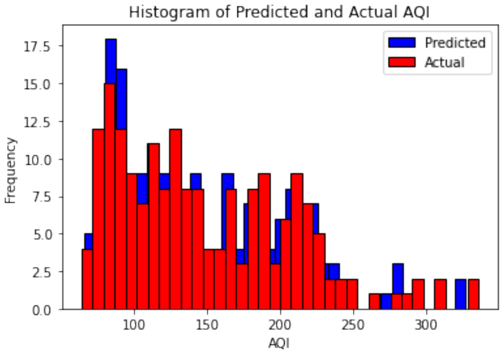


Figure 6.8: The Histogram of KNN For Dataset A

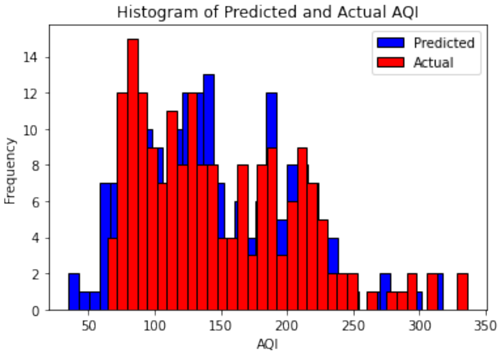


Figure 6.9: The Histogram of Lasso Regression For Dataset A

Dataset B

X axis = frequency
Y axis = AQI

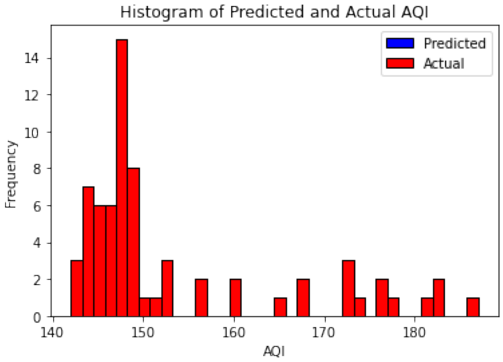


Figure 6.10: The Histogram of Linear Regression For Dataset B

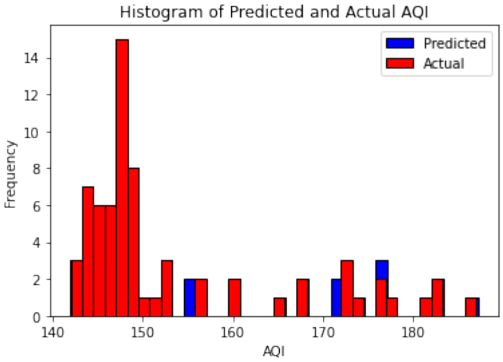


Figure 6.11: The Histogram of Random Forest Regression For Dataset B

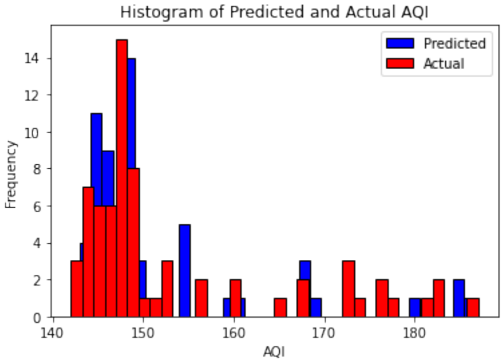


Figure 6.12: The Histogram of KNN For Dataset B

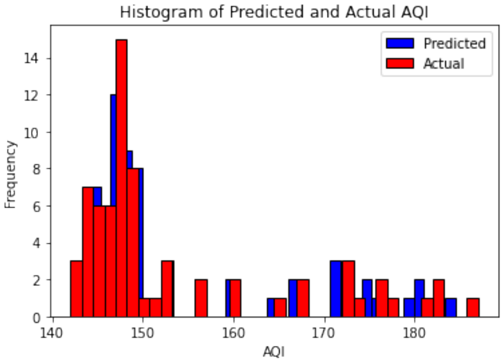


Figure 6.13: The Histogram of Lasso Regression For Dataset B

From the above histogram graphs it is noticeable that Linear Regression outperforms the other models.

6.2.2 Deep Learning Model Results

For single layer LSTM and double layer LSTM epochs=100 were taken.

For Single Layer Lstm

The loss and accuracy after every 1 epoch is plotted and showed below:

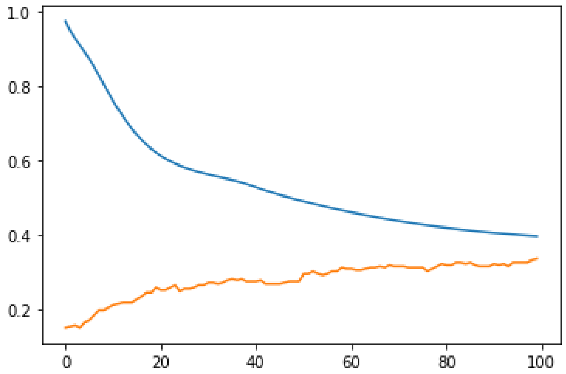


Figure 6.14: Loss and Accuracy for Single Layer LSTM

For Double Layer Lstm

The loss and accuracy after every 1 epoch is plotted and showed below:

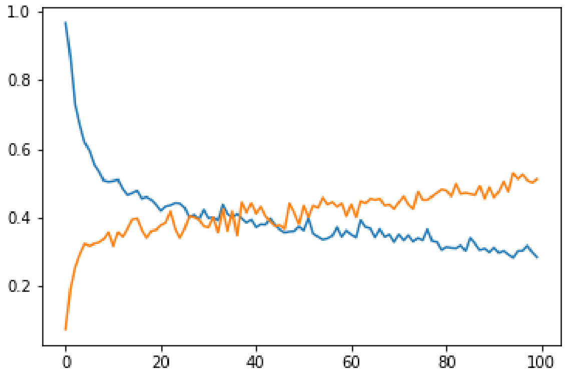


Figure 6.15: Loss and Accuracy for Double Layer LSTM

Deep Learning model was applied to Dataset B. For the Single Layer LSTM and Double Layer LSTM the train and test results are shown below:

Single Layer LSTM	**RMSE**
Train Score	3.80
Test Score	3.26

Table 6.3: Train and Test scores of Single Layer LSTM

Double Layer LSTM	**RMSE**
Train Score	2.65
Test Score	3.27

Table 6.4: Train and test scores of Double Layer LSTM

According to the result of the Single Layer LSTM's test score it can be seen that the test score is lower than the train score. Whereas, the Double Layer LSTM's test score is higher than the train score. So, from the test scores Single Layer LSTM worked better on the proposed dataset. Applying more layers to a neural network does not ensure that it will outperform simpler models on a given task. These are considered technical considerations (such as hyperparameters configuration) that one must test with the task and dataset to arrive at the best results.

Furthermore, by adding more components, such as the second LSTM layer that was decided to add, increases the model parameters that must be taught, which requires more time to train the layer. Whenever the number of model parameters increases, the model becomes more complex, making it difficult to fit on the training instances since it must enhance parameters in order to fit the training instances optimally.

Chapter 7

Limitations and Future Works

7.1 Limitations

There are several limitations to this study. None of which is a shortcoming to the success of this project. Through the allocation of time and resources, these are overcomable in the cear future.

1. In this research, we utilized a deep learning model, and in order to obtain the prediction, we require at least one year's worth of data. Due to the limited amount of time we had, we could only collect data for a brief period of time. As a consequence of it, the prediction of the model was not precise.

2. Calibration is required for all 3 mq sensors and the DHT11 sensor.

3. Power Failure isn't accounted for. Manual system initiation is required.

4. The system needs to be connected to a secure wifi.

5. Unavailability of a sensor that helps procuring the amount of SO2 present in Air. SO2 is one of the bigger pollutants alongside CO, NOx, and O3 in the city of Dhaka.

7.2 Future Works

To transfer the AQI data from the device to the server in the future, LoRaWAN (low-power, wide-area networking protocol) can be used. LoRaWAN sends short bits of data across greater distances several times each hour or day. As we will attempt to gather hourly data from our device, which will be located in a distinct range of locations from the server, LoRaWAN will be superior to Wi-Fi. It enables for minimal, low-data-rate transmission across a wide territory [28]. As we are employing IoT in our system, LoraWAN was designed exclusively for Internet of Things applications. However, this final category is being expanded with sensor networks, low power wide area networks, and a low 9-volt battery in the platform, which will last up to 10 years. We can utilize this facility to introduce DFL(Distributed Federated Learning) using multiple devices spread across a large area. The data that is to be gathered would be the trained model result of machine learning algorithms.

ThingSpeak, IBM, or any other comparable platform can be utilized to combine, visualize, and analyze live data streams on the cloud in the near future. The platforms can be used by collecting data on the air component on the Raspberry Pi while keeping track of the order so that the Raspberry Pi can supply concentration data in the right sequence. The ThingSpeak is available to the public, allowing anybody from any region to access the data in real-time. The channel is viewable in a browser as well as on Android smartphones with the app Thingsview. There are already hundreds of thousands of running similar projects on thingSpeak like this study. So there would be a better and fruitful data outcome as there would be multiple projects to verify data from.

GPS modules can be attached to the device and they can be assigned location identities. And these devices will cover one major residential/commercial area as a fleet of devices as a whole. At a later date, people will be able to access the data via a website or an app. So that individuals may receive the specific location's AQI hourly. As of now, daily or hourly AQI from various locations is not available in Bangladesh, and if it is accessible, it is expensive. They may view AQI updates as traffic jams on a map using Google API. Since it is the Covid-19 period and the air is densely polluted, germs and viruses may grow and spread diseases. As a result, individuals should avoid or be cautious while visiting polluted areas since the coronavirus may be present.

Jin [49] et. al. in their paper claimed that the best results are produced by MTMC - nLSTM while working with multivariate air quality forecasting. LSTM is the base stepping stone to modify, hypertune RNN algorithms for a accurate and better result for air quality prediction.

Chapter 8

Conclusion

Artificial intelligence is becoming the standard of data science research in the technological world by every passing moment. The better chance of gaining a proper set of organized data there is, the faster the gain of AI would be. The sole purpose of this research is to shift the gear of it's stride to the next level.

IoT is a part of every futuristic advancement the modern civilized world has made in the 21st century so far. This study hopes to open new doors in the further and near future by putting it's footprint in large scale research and creative endeavors.

Bibliography

[1] M. A. Poole and P. N. O'Farrell, "The assumptions of the linear regression model," *Transactions of the Institute of British Geographers*, pp. 145–158, 1971.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[3] A. Azad and T. Kitada, "Characteristics of the air pollution in the city of dhaka, bangladesh in winter," *Atmospheric Environment*, vol. 32, no. 11, pp. 1991–2005, 1998.

[4] K. A. Delin and S. P. Jackson, "Sensor web: A new instrument concept," in *Functional Integration of Opto-Electro-Mechanical Devices and Systems*, International Society for Optics and Photonics, vol. 4284, 2001, pp. 1–9.

[5] N. Habib, K. Mohammed, *et al.*, "Evaluation of planning options to alleviate traffic congestion and resulting air pollution in dhaka city," 2002.

[6] W.-Y. Chung and S.-J. Oh, "Remote monitoring system with wireless sensors module for room environment," *Sensors and Actuators B: Chemical*, vol. 113, no. 1, pp. 64–70, 2006.

[7] M. Kumar and M. Thenmozhi, "Forecasting stock index movement: A comparison of support vector machines and random forest," in *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[8] A. Venkatram, V. Isakov, E. Thoma, and R. Baldauf, "Analysis of air quality data near roadways using a dispersion model," *Atmospheric Environment*, vol. 41, no. 40, pp. 9481–9497, 2007.

[9] M. of Environment Forest. "Clean Air Sustainable Environment." (2011), [Online]. Available: http://case.doe.gov.bd/index.php?option=com_content&view=article&id=3&Itemid=18.

[10] R. Olyazadeh, *Geostatistic analysis, predicting the values of ozone concentration in the state of california*, Oct. 2012. DOI: 10.13140/RG.2.1.2924.6329.

[11] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 3, pp. 275–294, 2012.

[12] B. A. Begum, P. K. Hopke, and A. Markwitz, "Air pollution by fine particulate matter in bangladesh," *Atmospheric Pollution Research*, vol. 4, no. 1, pp. 75–86, 2013.

[13] S. C. Dogruparmak, G. A. Keskin, S. Yaman, and A. Alkan, "Using principal component analysis and fuzzy c–means clustering for the assessment of air quality monitoring," *Atmospheric Pollution Research*, vol. 5, no. 4, pp. 656–663, 2014.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] V. Rodriguez-Galiano, M. Sánchez Castillo, J. Dash, P. Atkinson, and J. Ojeda-Zujar, "Modelling interannual variation in the spring and autumn land surface phenology of the european forest," *Biogeosciences*, vol. 13, pp. 3305–3317, Jun. 2016. DOI: 10.5194/bg-13-3305-2016.

[16] R. Shete and S. Agrawal, "Iot based urban climate monitoring using raspberry pi," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 2008–2012. DOI: 10.1109/ICCSP.2016.7754526.

[17] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, "Raq–a random forest approach for predicting air quality in urban sensing systems," *Sensors*, vol. 16, no. 1, p. 86, 2016.

[18] K. Zheng, S. Zhao, Z. Yang, X. Xiong, and W. Xiang, "Design and implementation of lpwa-based air quality monitoring system," *IEEE Access*, vol. 4, pp. 3238–3245, 2016. DOI: 10.1109/ACCESS.2016.2582153.

[19] N. Desai and A. J.S.R., "Iot based air pollution monitoring and predictor system on beagle bone black.," *IEEE, 367–370.*, 2017. DOI: https://doi.org/10.1109/ICNETS2.2017.8067962.

[20] S. S. Ganesh, S. H. Modali, S. R. Palreddy, and P. Arulmozhivarman, "Forecasting air quality index using regression models: A case study on delhi and houston," in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, IEEE, 2017, pp. 248–254.

[21] M. Hansson, "On stock return prediction with lstm networks," 2017.

[22] C. V. Saikumar, M. Reji, and P. Kishoreraja, "Iot based air quality monitoring system," *International Journal of Pure and Applied Mathematics*, vol. 117, no. 9, pp. 53–57, 2017.

[23] Y. Song, J. Liang, J. Lu, and X. Zhao, "An efficient instance selection algorithm for k nearest neighbor regression," *Neurocomputing*, vol. 251, pp. 26–34, 2017.

[24] Components101. "Breadboard connections, features, examples and datasheet." (2018), [Online]. Available: https://components101.com/misc/breadboard-connections-uses-guide.

[25] ——, "Ds3231 rtc module pinout, configuration, example, circuit datasheet." (2018), [Online]. Available: https://components101.com/modules/ds3231-rtc-module-pinout-circuit-datasheet.

[26] C. Content team. "Diseases Caused By Air Pollution – Risk Factors and Control Methods." (2018), [Online]. Available: https://cleanair.camfil.us/2018/02/09/diseases-caused-by-air-pollution-risk-factors-and-control-methods/.

[27] S. Duangsuwan, A. Takarn, and P. Jamjareegulgarn, "A development on air pollution detection sensors based on nb-iot network for smart cities," in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, IEEE, 2018, pp. 313–317.

[28] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of lorawan for iot: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, 2018.

[29] A. Holzinger, "From machine learning to explainable ai," in *2018 world symposium on digital intelligence for systems and machines (DISA)*, IEEE, 2018, pp. 55–66.

[30] T. d. S. Staff Correspondent. "Pollution the killer." (2018), [Online]. Available: https://www.thedailystar.net/environment/environment-pollution-in-dhaka-bangladesh-18000-died-world-bank-report-1634566.

[31] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, and M. Daneshmand, "Internet of things mobile–air pollution monitoring system (iot-mobair)," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5577–5584, 2019.

[32] M. Korunoski, B. R. Stojkoska, and K. Trivodaliev, "Internet of things solution for intelligent air pollution prediction and visualization," in *IEEE EUROCON 2019-18th International Conference on Smart Technologies*, IEEE, 2019, pp. 1–6.

[33] MisterBotBreak. "How to use a real-time clock module (ds3231)." (2019), [Online]. Available: https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-real-time-clock-module-ds3231-bc90fe.

[34] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[35] A. L. Association. "Nitrogen dioxide." (2020), [Online]. Available: https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/nitrogen-dioxide.

[36] ——, "Sulfur dioxide." (2020), [Online]. Available: https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/sulfur-dioxide.

[37] V. K. R. Chimmula and L. Zhang, "Time series forecasting of covid-19 transmission in canada using lstm networks," *Chaos, Solitons Fractals*, vol. 135, p. 109 864, 2020, ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2020.109864. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920302642.

[38] E. Hossain, M. Shariff, M. Hossain, and K. Andersson, "A novel deep learning approach to predict air quality index," in Dec. 2020, pp. 367–381, ISBN: 978-981-33-4673-4. DOI: 10.1007/978-981-33-4673-4_29.

[39] J. A. Moscoso-López, D. Urda, J. González-Enrique, J. J. Ruiz-Aguilar, and I. J. Turias, "Hourly air quality index (aqi) forecasting using machine learning methods," in *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, Springer, 2020, pp. 123–132.

[40] L. Moses *et al.*, "Iot enabled environmental air pollution monitoring and rerouting system using machine learning algorithms," vol. 955, no. 1, p. 012 005, 2020. DOI: https://doi.org/10.1088/1757-899x/955/1/012005.

[41] E. Mussumeci and F. C. Coelho, "Machine-learning forecasting for dengue epidemics - comparing lstm, random forest and lasso regression," *medRxiv*, 2020. DOI: 10.1101/2020.01.23.20018556. eprint: https://www.medrxiv.org/content/early/2020/01/24/2020.01.23.20018556.full.pdf. [Online]. Available: https://www.medrxiv.org/content/early/2020/01/24/2020.01.23.20018556.

[42] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios," *Ieee Access*, vol. 8, pp. 23 022–23 040, 2020.

[43] D. Smith. "A guide to understanding particulate matter (pm)." (2020), [Online]. Available: https://learn.kaiterra.com/en/air-academy/particulate-matter-pm.

[44] X. Zhao, M. Song, A. Liu, Y. Wang, T. Wang, and J. Cao, "Data-driven temporal-spatial model for the prediction of aqi in nanjin," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, 2020.

[45] P. Chhikara, R. Tekchandani, N. Kumar, M. Guizani, and M. M. Hassan, "Federated learning and autonomous uavs for hazardous zone detection and aqi prediction in iot environment," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 456–15 467, 2021. DOI: 10.1109/JIOT.2021.3074523.

[46] Components101. "Arduino uno pinout, specifications, pin configuration and programming." (2021), [Online]. Available: https://www.components101.com/microcontrollers/arduino-uno.

[47] ——, "Dht11 sensor pinout, features, equivalents and datasheet." (2021), [Online]. Available: https://components101.com/sensors/dht11-temperature-sensor.

[48] EPA. "Ground-level ozone basics." (2021), [Online]. Available: https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics.

[49] N. Jin, Y. Zeng, K. Yan, and Z. Ji, "Multivariate air quality forecasting with nested long short term memory neural network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8514–8522, 2021.

[50] F. Nawazi. "Mq-131 ozone gas sensor module." (2021), [Online]. Available: https://robu.in/product/mq-7-co-carbon-monoxide-coal-gas-sensor-module/.

[51] ——, "Mq-135 air quality smoke gas sensor module." (2021), [Online]. Available: https://robu.in/product/mq-7-co-carbon-monoxide-coal-gas-sensor-module/.

[52] S. Sharma. "What is air quality index (aqi) how is it calculated ?" (2021), [Online]. Available: https://www.pranaair.com/blog/what-is-air-quality-index-aqi-and-its-calculation/.

[53] Z. Zhu, Y. Qiao, Q. Liu, *et al.*, "The impact of meteorological conditions on air quality index under different urbanization gradients: A case from taipei," *Environment, Development and Sustainability*, vol. 23, no. 3, pp. 3994–4010, 2021.

[54] Airnow. "Air Quality Index (AQI) Basics." (2022), [Online]. Available: https://www.airnow.gov/aqi/aqi-basics.

[55] T. W. B. in Bangladesh. "Overview." (2022), [Online]. Available: https://www.worldbank.org/en/country/bangladesh/overview#1.

[56] C. A. R. Board. "Inhalable particulate matter and health (pm2.5 and pm10) — california air resources board." (2022), [Online]. Available: https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health.

[57] EPA. "Epa." (2022), [Online]. Available: https://www.epa.gov/.

[58] A. S. Gillis. "What is the internet of things (IoT)?" (2022), [Online]. Available: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT.

[59] IQAir. "Air Quality Analysis and Statistics For Bangladesh." (2022), [Online]. Available: https://www.iqair.com/bangladesh.

[60] U. Lokhande. "Introduction to raspberry pi 3 model b." (2022), [Online]. Available: https://binaryupdates.com/introduction-of-raspberry-pi-3-model-b/.

[61] M. A.-M. Molla. "Air pollution takes 3 years off life in Bangladesh." (2022), [Online]. Available: https://www.thedailystar.net/environment/pollution/air-pollution/news/3-years-chopped-2974306.

[62] U. N. .-. W. P. Prospects. "Dhaka, Bangladesh Metro Area Population 1950-2022." (2022), [Online]. Available: https://www.macrotrends.net/cities/20119/dhaka/population.

[63] W. P. Review. "Dhaka Population 22." (2022), [Online]. Available: https://worldpopulationreview.com/world-cities/dhaka-population.

[64] S. N. Sakib. "More residents developing respiratory illnesses due to polluted air in Bangladeshi capital." (2022), [Online]. Available: https://www.aa.com.tr/en/asia-pacific/more-residents-developing-respiratory-illnesses-due-to-polluted-air-in-bangladeshi-capital/2520591.

[65] T. W. A. Q. I. project, "Dhaka us consulate, bangladesh air pollution: Real-time air quality index," *aqicn.org*, [Online]. Available: https://aqicn.org/city/bangladesh/dhaka/us-consulate?fbclid=IwAR0-bfO1TMBFlcTSsiwoXFJrAepIwqw1uyiy3h_5QXsMv1QnndR3I4WwLdo.

[66] K. Electronics. "Pm2.5 air quality sensor and adapter kit - pms5003." (—-), [Online]. Available: https://www.kiwi-electronics.nl/en/pm2-5-air-quality-sensor-breadboard-adapter-kit-pms5003-3398.

[67] R. incorporated. "Mq-7 gas sensor user manual." (—-), [Online]. Available: https://robu.in/wp-content/uploads/2021/08/MQ-7-Gas-Sensor-UserManual.pdf.

[68] C. Marketplace. "Mq135 air quality sensor module." (—-), [Online]. Available: https://sg.cytron.io/c-sensor/c-gas-sensor/p-mq135-air-quality-sensor-module.

[69] PriyoShop. “Transcend 64gb uhs – i u1 micro sd memory card ts64gusd300s–a.” (—-), [Online]. Available: Transcend%2064GB%20UHS%20%E2%80%93%20I%20U1%20Micro%20SD%20Memory%20Card%20TS64GUSD300S%E2%80%93A,%20PriyoShop.com%20-%20Online%20Shopping%20in%20Bangladesh.

[70] Robu.in. “Buy carbon monoxide mq7 gas sensor module.” (—-), [Online]. Available: https://robu.in/product/mq-7-co-carbon-monoxide-coal-gas-sensor-module/.

% Where the bibliography will be printed