

A Novel Approach to Reduce Air Pollution Through Machine Learning Based PM2.5 Prediction.

by

Omar Farhad Alif
18101167

Tarik Monwar Monsaif
18101172

Swakshar Das Amarth
18101157

Md. Nafiu Kabir
18101440

Tahmid Asif Sadman
18101166

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

© 2022. Brac University
All rights reserved.

Declaration

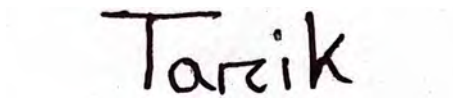
It is hereby declared that

1. The thesis submitted is 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:



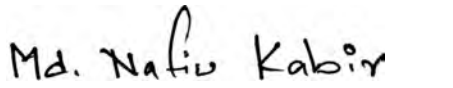
Omar Farhad Alif
18101167



Tarik Monwar Monsaif
18101172



Swakshar Das Amarth
18101157



Md. Nafiu Kabir
18101440



Tahmid Asif Sadman
18101166

Approval

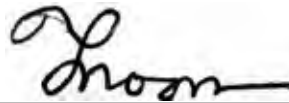
The thesis titled “A Novel Approach to Reduce Air Pollution Through Machine Learning Based PM2.5 Prediction” submitted by

1. Omar Farhad Alif (18101167)
2. Tarik Monwar Monsaif (18101172)
3. Swakshar Das Amarth (18101157)
4. Md. Nafiu Kabir (18101440)
5. Tahmid Asif Sadman (18101166)

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

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

AI (Artificial Intelligence) is one of the most fascinating subjects for computer scientists in this century. It is a technology that helps a machine to do tasks by itself which needs human intelligence. This technology reduces human error and increases the productivity of any task by a greater number of times. Nowadays AI is being used by many devices and also it has a limitless possibility to work with. As this technology grows faster scientists are combining it with various other technologies to be perfect. Cloud computing and IoT (Internet of Things) can be used in a framework for an AI to be more intelligent and give access to more data. For the past few years, only a small group of people around the world are thinking about the environment that they are living in where the environment must be the priority to think about. Air pollution is one of the causes which people are gradually becoming sick. Polluted air causes different lung diseases for which people also need to bear medical expenses. When technology is in the hands of every person, it is a must to make them aware of what they are inhaling and how they can make the world a better place. So, the answer to the question of what we are doing is that this research proposes a novel approach for reducing air pollution by making people more aware of their day-to-day life by detecting the environmental air condition with the help of machine learning by using PM2.5 prediction and computational cloud computing. As a result, we will introduce a framework that allows one to know about the air that one is inhaling and how to counter the air pollution by providing a computational output with the help of PM2.5 prediction, Computational Cloud Computing, Networking, and IoT with necessary actions with the help of some known methods such as Logistic Regression Analytics, Artificial Neural Networks, Decision Tree and Long short-term memory neural networks.

Keywords: Artificial Intelligence; Machine Learning; Computational Cloud Computing; reducing air pollution; Prediction; Decision tree; Linear Regression Analysis.

Dedication

Firstly, this project is dedicated to our almighty creator, the source of all knowledge, power, and understanding. He has given us the human the power to change many things. This power should be used for the betterment of this world. In the name of the almighty, we tried to use our knowledge as much as possible for doing something better which can contribute to the betterment of future generations.

Secondly, this thesis is also dedicated to the future generation, who will further work on this project we believe. Hope you guys will discover many other things which we could not because of the lack of time and technology of our time. Hope you guys will work with the intention of doing something great rather than bookish knowledge. God bless.

Acknowledgement

Firstly, all the praise goes to the almighty creator. He is the supreme one and the source of all power and knowledge. Without his grace, we could not have done this properly without any interruption.

Secondly, thanks to our supervisor Jannatun Noor Mukta ma'am for her advice and constant support throughout the time. She was very friendly and always had a positive attitude towards our work. Also, she tried to push us to our limits as a result we were able to come up with good results. She was always there for us regarding all the facts.

Last but not the least, thanks to our family members for their constant mental support and inspiring us whenever we felt depressed.

God bless all.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	xi
Nomenclature	xii
1 Introduction	1
1.1 Research Problem	2
1.2 Research Objectives	4
1.3 Our Contributions	5
1.4 Thesis Outline	5
2 Literature Review	6
2.1 AI-assisted framework	6
2.2 PM2.5 gets reduced by Rain & Snow	6
2.3 Cloud Computing	6
2.4 Related Works	6
3 Methodology	9
3.1 Input Data	12
3.1.1 PM2.5 detection Device	14
3.1.2 PMS5003 Air Quality Sensor	14
3.1.3 Interfacing PMS5003 PM2.5 Air Quality Sensor with Arduino	16
3.1.4 Source Code	16
3.1.5 Interfacing PMS5003 PM2.5 Air Quality Sensor with Arduino LCD Display	18
3.1.6 Source Code for LCD	18
3.1.7 Storing Data	19

3.2	Data Pre-processing	20
4	Implementations and Result	22
4.1	Implementation	22
4.1.1	Logistic Regression	22
4.1.2	Decision Tree	23
4.1.3	GaussianNB Algorithm	24
4.1.4	Gradient Boosting Algorithm	25
4.1.5	SVC Algorithm	26
4.1.6	Passive Aggressive Classifier	27
4.1.7	SGD Algorithm	28
4.1.8	MLP Classifier Algorithm	29
4.1.9	KNN Algorithm	30
4.1.10	Linear SVC	31
4.1.11	Random Forest Algorithm	32
4.2	Confusion Matrix	34
4.3	Prediction	35
4.3.1	Linear Regression	35
4.4	Input Data Pre-pressing	37
4.5	Result implementation	37
5	Limitations and Future work	41
6	Conclusion	42
	Bibliography	45

List of Figures

1.1	AI for Clean Air	2
3.1	AI for Clean Air	9
3.2	The flow chart of the proposed model for reducing air pollution . . .	11
3.3	Air quality image worldwide [28]	13
3.4	Air quality and PM2.5 pollution in Dhaka [28]	13
3.5	AQI Category and Health Implications [9]	13
3.6	AI for Clean Air [22].	15
3.7	PM2.5 AQ sensor connected with Arduino [22].	16
3.8	Real-time PM2.5 AQ sensor connected with Arduino [22].	16
3.9	Source code	17
3.10	Interfacing PMS5003 PM2.5 with Arduino LCD Display	18
3.11	We are getting data using the device	18
3.12	LCD source code	19
3.13	Categorical Value Conversion	20
3.14	Input data before and after pre-processing	21
4.1	Logistic Regression [2]	22
4.2	Logistic Regression Accuracy Test.	23
4.3	Decision Tree [7]	23
4.4	Decision Tree Accuracy Test.	24
4.5	Gaussian Naïve Bayes Algorithm [18]	25
4.6	Gaussian Naïve Bayes Algorithm Accuracy Test.	25
4.7	Gradient Boosting Algorithm Accuracy Test	26
4.8	SVC Algorithm Accuracy Test.	27
4.9	Passive Aggressive Classifier Accuracy Test.	28
4.10	SGD Algorithm [30]	29
4.11	SGD Algorithm Accuracy Test.	29
4.12	MLP Classifier Accuracy Test.	30
4.13	K Neighbors Classifier Algorithm Accuracy Test	31
4.14	Linear SVC Algorithm Accuracy Test.	32
4.15	Random Forest Classifier Algorithm [19]	32
4.16	Random Forest Classifier Algorithm Accuracy Test.	34
4.17	Confusion Matrix	34
4.18	Linear Regression.	36
4.19	A Sample of Raw Data of the collected Dataset.	37
4.20	Accuracy Graph.	38
4.21	Linear Regression Algorithm Prediction Values.	39
4.22	Circuit model	39

4.23 Device model	40
-----------------------------	----

List of Tables

3.1	Table of the components description	14
3.2	Description of functions.	15
4.1	Accuracy of Algorithms.	38

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

AQI Air Quality Index

GND Ground

IOT Internet of Things

LSTM Long Short-Term Memory

OTCR Office of Computational and Technical Research

PM2.5 Particulate Matter 2.5

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SO2 Sulfur dioxide

SVM Support Vector Machine

TTL Transistor-transistor logic

UART Universal Asynchronous Receiver-Transmitter

VCC Voltage Common Collector

WHO World Health Organization

WRFCHEMMODEL Weather Research and Forecast model coupled with Chemistry

WSN Wireless Sensor Networks

Chapter 1

Introduction

Since the beginning of the industrial revolution, air pollution has increased day by day not only because of industrial wastage but also for people cutting down trees, burning plastics, carbon emission, and last but not least for lack of awareness. If people were aware of the air that they are inhaling every moment, they would be more careful before harming the environment and would also reduce the amount of air pollution for a better future.

Now, as the world is changing day by day rapidly with the help of technology, it is wise to take full advantage of technology for the betterment of mankind. As we already know that, AI is the future of human intelligence because the things that can not be predicted by the human can be predicted with the help of machine learning very easily. Machine learning is the latest technology that has the potential to change the world. Machine learning is consisting of two-component which are artificial and intelligence. Artificial means something made or produced via way of means of humans instead of going on naturally, mainly reproduction of something natural. In the same way, intelligence means the ability to acquire skill or knowledge by own. Artificial intelligence means a machine or a system that is created by humans will be able to learn things on its own.

As the type of pollution on the air is changing day by day the system needs to adapt to this change and provide a realistic solution accordingly. For this purpose, machine can collect data throughout the internet with the help of cloud computing and provide an optimal solution that humans can apply to reduce air pollution by a lot. In one of the articles published by the Department of Civil Engineering, Jamia Millia Islamia University, New Delhi says that from 2003 to 2021, the work based on AI for detection and reducing air pollution has grown rapidly as the AI-based techniques have emerged as the maximum effective and forward-searching procedures for air pollutants forecasting due to their unique capabilities consisting of natural learning, excessive precision, advanced generalization, robust fault tolerance and simplicity of operating with multi-dimensional data [23].

AI is used by almost every high-tech system such as manufacturing robots, self-driving cars, smart assistance, proactive healthcare management, disease mapping,

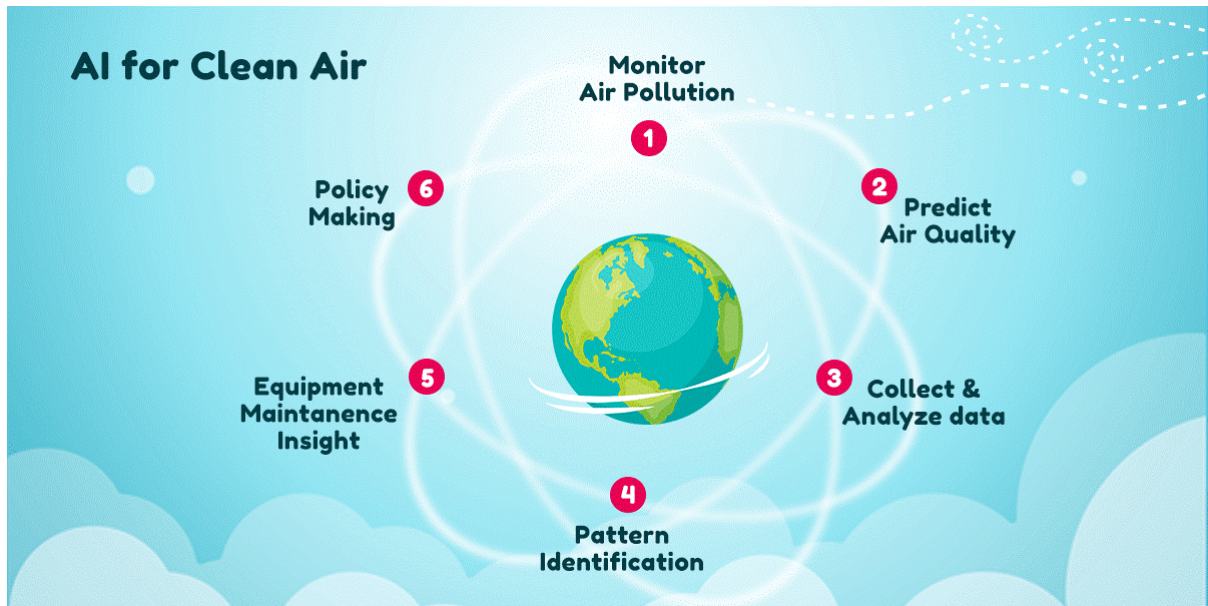


Figure 1.1: AI for Clean Air

automatic financial investing, virtual travel booking agent, social media monitoring, intern team chat tool, conversational marketing bot, natural language processing tools and many more. In contrast, rather than using it in high tech industries, our focus should be using AI in our daily life for not only making it as an industrial use but for the people to take full advantage of this technology.

1.1 Research Problem

As the world is rapidly preparing for the fourth industrial revolution, the quantity of pollutants in the air is increasing day by day. According to [1], research conducted by gems shows that, the pollutant level of SO₂ and SVM is still higher than the marginal scale which is provided by WHO and it still needs to be maintained. This refers to all urban areas, which fall under the section of massively polluted areas. However, they also predicted that, if all the countries' data are included in their research this number could be much larger than the number they have got. According to [4], global air pollution only occurs because of the pollutants which have a lifetime of more than 1 week. This time of pollution can be determined and monitored by cloud-centric systems which give us a real-time view of the pollution spikes. However, this type of system is inefficient. According to [24], exposure to air pollution spikes can harm our bodies greatly. However, as the cloud-centric system is not enough efficient there is also an edge-centric system that allows us to monitor the pollution spikes more accurately with the help of machine learning by comparing them with the pollution limits defined by the standards.

Cloud-centric systems work within the cloud by running the workload in it. On the other hand, edge-centric systems run the workload on edge devices. Regardless, the inefficiency that the cloud-centric system shows, it can be accessed from all over the internet but for the case of an edge-centric system, it can only take the workload and can only be accessed by the edge devices [24].

Nowadays, machine learning system is only used for detecting air pollution and to

measure pollutant quantities. However, it cannot reduce air pollution if human does not take the matter into their hand. This technology can only give us instructions by calculating the odds for reducing air pollution but we have to follow those instructions and do the necessary things to reduce this damage.

There are so many approaches to detecting air pollution through the cloud. However, using the machine learning approach is the most effective way on the cloud-centric system. According to [12], some models can be followed to detect air pollution such as Linear Regression Analytics, Artificial Neural Networks, and Long short-term memory neural networks which all are cloud-centric. Yet, it is often challenging to say which one is the best because the problem at hand is always changing and there are also some advantages and disadvantages for all of the approaches.

According to [12], Linear Regression Analytics is a classic approach to model the relationship between a variable, which corresponds to features and results in a given dataset. This approach has some pros and cons. This approach shows a generally acceptable RMSE but this model cannot predict high PM2.5 values. PM2.5 means Fine Particulate Matter which has 2.5 diameters of value or less than this. On the other hand, according to [12], an Artificial Neural Network is also called ANN or perceptron which is a connection system used to simulate biological neural networks. As a result, they can learn from the information in a much common manner than humans. In the article [12], it is said by using an Artificial Neural Network (ANN) one can get a more accurate result as it has a good performance for describing any linear or non-linear relationship that may exist in a dataset. Which can be used diversely in many research areas. However, dependency on high resource consumption is a drawback of using this approach. And it only works well against a given dataset. Last but not least, Long Short-Term Memory (LSTM) Neural Network is also known as Recurrent Neural Network (RNN). According to [12], it was firstly proposed in 1997 to solve the problem with traditional RNNs, where the information could be preserved in long sequences. In RNN, it reads the sequence of samples as an input value where the output can be both a sequence or a single value. The LSTM model has far less both in general and for high values but the training process of LSTM can be very time-consuming hence there is a need for large-scale computing resources. Also, to make this model more effective it needs a large amount of data and the collection can be a much longer process.

As the internet has become more accessible nowadays, there are tons of data on air pollution. Even authentic sources like google are making a step to telecast live pollution rates in areas around the world through their satellite. They also provide images and graphs of air pollution by which the pollution rate can be read and processed by image processing. As all the data in the computer is read as binary, the images can be processed by AI by using the approaches that we talked about. According to [3], by observing the surface of the air of an area and also by detecting the concentration of particle air pollution can be detected by a machine by using several mathematical formulas and also by observing dimensional interpolation. Also, by analyzing the data from the picture it is possible to image segmentation, feature extraction, classification, and detection of the most important sources of pollution, prediction, and control of pollution sources.

However, the problem is that using machine learning there are so many datasets to consider and there are so many variants of air pollution with different types of particles which we need to point out. According to [29], there are several types of

air pollution but the most common of them is particulate matter, PM2.5, sulfur dioxide, ozone, and nitrogen dioxide. Even there are organic and chemical variants among them. As the data set can only show the pollutant and pollution rate it is hard to detect with machine learning which type of pollutant is polluting the air daily. As machine learning can gather datasets and pieces of information and also can give us an optimal solution, we need to act on that solution also.

Now, the question is that, whether machine learning technology can reduce air pollution rather than only detecting it. The answer is yes because lots of researches have been done in this area and also more researches are being done nowadays. But there is also another question, whether machine learning can reduce the air pollution worldwide environmentally rather than of a small space and how effective can it be in the future. Therefore, the question that the research will answer:

RQ.1 “Can machine learning be a solution for reducing air pollution by processing images from live maps and giving humans optimal instructions to act upon them?”

and also, if it is possible then,

RQ.2 “Is there any better module that can be followed to fulfill this?”

As mentioned earlier, different approaches can be followed which are [12], Logistic Regression Analytics, linear regression Artificial Neural Network, Decision Tree, and Long Short-Term Memory. Also, the approaches have several advantages and disadvantages.

This research will answer the above questions and also find out the better way of using machine learning for reducing air pollution for a better environment.

1.2 Research Objectives

This research focuses on developing a system that will not only detect PM2.5 particles in the air by help of machine learning but also give a countermeasure that can be performed instantly or can be performed later to reduce air pollution. The main goal of this system will be to provide AI support hand to hand by sending notifications via different devices to make the user aware of his surroundings. As an example, if an area is heavily polluted with PM2.5 particles then the system will alert the user with the help of AI to act on the countermeasure. On a large scale, by processing the PM2.5 concentration in the air of an area through a PM2.5 detection device which will predict an optimal concentration of PM2.5 and give instructions that need to be followed by the user to reduce the pollution of that area.

As a result, the main objective is this research is to:

1. Understand AI, machine learning and Cloud Computing properly and how it works.
2. Understand the detection process and prediction of PM2.5 concentration in the air.
3. Come forward with an optimal approach for AI to work properly.

4. Develop a framework for the proposed topic.
5. Evaluate the framework.
6. Find the drawback of the framework.
7. Improve the framework.

1.3 Our Contributions

We figured out an optimal algorithm for PM2.5 prediction. We have predicted PM2.5 concentration by using linear regression. We have created a PM2.5 detection device that can be upgraded in future. We have given a primary solution to decrease PM2.5 concentration from the air.

1.4 Thesis Outline

In the introduction part, the problem we will solve through this research has been discussed. Moreover, our research objectives have been declared thoroughly. Our contributions towards our paper have also been added. In the literature review part, the topics which is related to our mentioned theme has been highlighted. In the methodology part, the creation of PM2.5 detection device has been explained. Not only that, how the device will store data and how it will pre-process is also presented. In the implementation part, all the algorithms that we have used to figure out the accuracy with the code and confusion matrix has been added. The prediction for PM2.5 is also highlighted in this section. In limitations and future work, the issues that hindered our project has been discussed. But at the same time how we can improve in the future also addressed in the part. We concluded by creating a picture of Air Pollution in the current world and the importance of reduction of this. We have also used several appropriate references throughout the paper.

Chapter 2

Literature Review

2.1 AI-assisted framework

There are various reasons for using AI-assisted framework on projects in the real world and one of them is optimization and durability. Moreover, AI-assisted frameworks are cheap compared to other frameworks and faster which gives them additional advantages.

2.2 PM2.5 gets reduced by Rain & Snow

According to [26] as a consequence of the rain and snowy environment increasing root outflows and decreasing atmospheric convection, the concentration of PM2.5 is reduced, which improves air quality. The rain clears out PM2.5 from the air. Then because of gravity, the particles settle down on the ground. Furthermore, rainfall can affect the PM2.5 concentration for a period of time. It might happen because rain leads to atmospheric dampness. In this way, particles collide with raindrop fusion and deposition since rainfall happens alongside strong winds.

2.3 Cloud Computing

Cloud computing is the shipping of computing offerings over the internet. It delivers on-demand processing power, storage, and applications. As we keep our data in computer hard drives which takes a lot of space cloud to provide a remote database to keep data, access, and modify them. So, in this modern world cloud computing is an effective way to access data over one platform between users.

2.4 Related Works

In this section, we will have a look at some of the related work which has been done in the field of air pollution with the help of an AI-assisted framework, high-quality image processing, and computational cloud computing.

A lot of discussions have already taken place in every part of the world on how Cloud Computing can contribute to solving one of the major problems of the current world,

Air Pollution. A lot of practice has also taken place on the matter. But this particular paper [10] points out one issue in the ongoing practices of using cloud computing to monitor air pollution. That is managing the cloud resources. This paper [10] addresses that the algorithms which have been used so far take a lot of unnecessary resources thus overload the hosts. Which results in costly maintenance. So, this paper [10] comes out with a solution about how the resources can be managed properly get the best out of them. This paper [10] comes out with an algorithm called "OTCR" to address this problem. This algorithm will sort out the resources and distribute them to hosts according to their capability their current situation. The primary goal of this algorithm is to not let any of the hosts get overloaded or underloaded with resources. Whenever a task reaches the host the algorithm with works to seek out the suitable host which can utilize the resource best. If all the hosts are overloaded, it'll then scan get the list of all the live tasks and then find out the task with the least resources and then will switch it. The algorithm would work similarly in case of any host being underloaded with resources. This paper [10] would help the usage of Cloud Computing in monitoring air pollution by sharpening it by making the best usage of the resources. Though one drawback of the paper [10] is it doesn't mention how to drop out the unnecessary resources. By dropping them out the management of the resources could've been better.

For our health, predicting the air pollution of our surroundings is very important. This research work [12] emphasizes the fact that traditional air pollution prediction methods have limitations but the research has also provided us with a very effective solution to the problem which is the use of different machine learning algorithms. The machine learning approaches they used are linear regressing analytics, artificial neural networks, and long and short-term memory analytics. In the paper [12], they also showed the good and bad sides of each of the models. Moreover, in this paper, they combined different pollution-related data sets to analyze the data and applied those machine learning approaches to them. Furthermore, the Nectar research cloud is used to train artificial neural networks, long and short-term memory analytics models.

In urban cities air pollution is increasing day by day and predicting how much air pollution is occurring is very important to manage and reduce it. According to [8] a model named WRF CHEM MODEL (Weather Research and Forecast model coupled with Chemistry) is used to simulate air pollution, but it has some disadvantages: the source list is not updated in time. In this paper [8] , they used the full advantage of WRF CHEM MODEL for features and inputs but to improve the prediction performance they designed an evolution framework of AI. In this paper [8] , the experiment is done on different cities of China to get the maximum result from their data in the model.

For a pleasant environment, it is necessary to visualize the air pollution as estimates in 2018 disclose that 9 out of 10 people breathe air containing high levels of pollutants [11]. According to WHO, outdoor and indoor air pollution is responsible for about 7 million deaths globally per year. It is necessary to make 2D and 3D maps by using the Geographical Information System which is very time-consuming and requires a lot of data. In this paper [6] cloud computing is used along with

Geographical Information System to reduce the visualization time. Along with that Smart-MapReduce is used for help and to make the 3D air pollution maps which provide a very satisfactory result.

According to [13] it's far stated that real-time air pollutants tracking structures Using Wireless Sensor Networks Connected in a Cloud-Computing are a Wrapped-up Web Service. Air pollutants are affecting our surroundings and reducing our best of existence. For making sure the surroundings in which we and our youngsters can stay properly, we want to layout and put in force a steady and low-price real-time air pollutants tracking device. According to [13], implementation of a real-time Air Pollution Monitoring System primarily based totally on the usage of WSN below the idea of IoT the usage of the infrastructure of Cloud Computing may be a solution. A three-layer structure may be designed and carried out with low-price digital hardware, a Web carrier also can be designed and carried out the usage of a fixed of protocols and codecs used to system the records and keep them in a database as part of the Cloud infrastructure. The WSN and customers may be capable of talking thru a net-primarily based graphical consumer interface. According to [13], the clinical network has exhibited diverse answers primarily based totally on traditional Wireless Sensor Networks (WSN) for air pollutants tracking. Due to a loss of representing low-price answers, a few require hiring web website hosting or net carrier. To upload to that, we would have numerous constrained messages without a fallback method. The purpose of this painting is to expand a confident environmental tracking device primarily based totally on WSN this is incorporated into the Internet of Things (IoT) idea which will increase the theory and existence span of the sensor nodes of the WSN tremendously low costs. Firstly, each hardware and software program prototype must be blended with the usage of Arduino and Raspberry Pi structures which incorporates more than one air pollutants sensor in addition to newly built wi-fi enlargement modules[27]. Secondly, a three-layer structure must leverage a real-time air pollutant tracking device that has been deliberate and carried out. The first sensor layer subsumes the digital hardware circuits and the software program components, each for the Arduino-primarily based sensor nodes and the gateway node. This has been protected as it becomes assembled the usage of a Raspberry Pi collectively with a low-price wi-fi enlargement module for taking pictures of the records. Web carrier has been designed and carried out the usage of a fixed of protocols and codecs which might be used to system the records and keep them in a database as a part of the Cloud infrastructure. The purchaser layer will encompass a Web graphical consumer interface on the way to offer visible records of approximately environmental parameters so that it can permit communicate with the WSN and customers.

Chapter 3

Methodology

To execute a work properly a better execution plan is always needed. The proposed framework works more or less in the same way. The model first takes data from the input and then reads the data. After reading the data it processes the data with the help of AI. AI helps to process the image by segmentation, feature extraction, classification, and detection After processing is finished the data goes through some algorithm. This algorithm can differ from various conditions such as workload, time consumption, computing status, etc. By going through this process AI detects whether the air is polluted or not. If the air is polluted then it gives an optimal solution that is needed to be executed by humans and if not then it goes back to data processing.

Firstly, a system can be built which will detect air pollution levels in the air, and then it will use an air detection sensor to detect the pollution in that particular environment. The sensor can be a PM2.5 sensor that will monitor and measure the air quality. PMS5003 is such kind of a sensor [14] that is used to get the quality and quantity in the unit volume of suspended particulate matter and a digital interface in form of output. The sensor will detect the rate of pollution in that environment at different times of the year.

It can also be embedded in a variety of concentrations of suspended particulate mat-



Figure 3.1: AI for Clean Air

ter in the air or related instrumentation equipment to improve the environment. Not to mention, it also provides timely and accurate concentration data. In an industrial area where the pollution rate is very high, in that environment, we can use any kind of sprayer which will react by reading the data from the sensor and spray counter substance according to the pollutant substance. The device can be very small and cheap so anyone can use it in the socio-economic condition of Bangladesh.

Secondly, when we get a response from the sensor for a certain area, we will know that the air of that area is polluted. According to [17], certain trees produce oxygen 24 hours and purify the air such as Aloe Vera, Neem, Peepal, Tulsi, and Money Plant. If it is possible to calculate the right number of trees that needed to be planted in the area for reducing air pollution initiatives can be taken but this is still a question that needed to find out.

Figure 3.2 provides a high-level view of the model design.

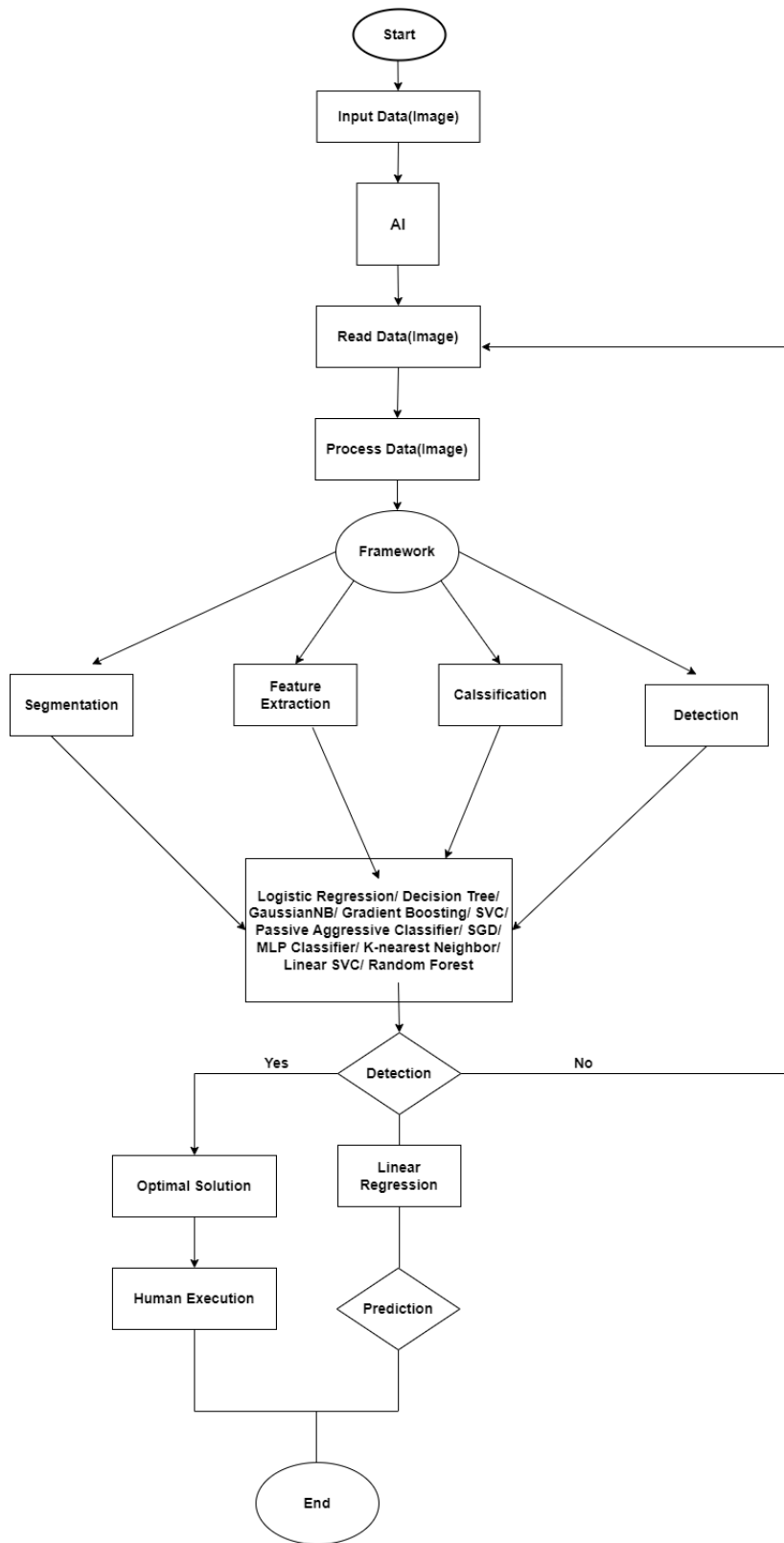


Figure 3.2: The flow chart of the proposed model for reducing air pollution

Logistic regression, Artificial Neural networks, Decision Trees, and Long Short-Term Memory which are also known as RNN algorithms are used to detect air pollutants from the air.

We will add an extra feature where the framework will give a tree count of a particular area. As we know, tree acts as a filter for making the environment pollution-free. Here The framework will give a tree count of an area where it will also say how many trees needed to be planted for reducing the air pollution of the area as a counter effect.

Here the challenge is to combine image processing and AI-based algorithm to not only detect air pollution but also to provide an optimal solution that can be implemented in real life.

3.1 Input Data

An authentic and reliable data set is a must for achieving an authentic result. So, it is very important to find a reliable authentic data set with the collection of various attributes and sources. As the main theme of our work is to process images from the internet and give a reading on the air pollution with sustainable feedback firstly, we need to find a source from where we can take the image and process it for a minimal result. Now, to talk about image processing data, taking data from all over the world is not possible. According to [28], we can see that the live air quality map shows us the air quality all over the world and we can say that, in this stage of our work a big data set like this can be a threat in the way of minimal result. For now, we have taken a data set that gives us the reading of PM2.5 particle that was present in the air from 2017 to 2022. This data is updated month to date and can be collected from¹

Moreover, for the reading of PM2.5 values image processing is needed. In figure 3.5 we can see that the value of the PM2.5 particle is determined with some colors which will be read through image processing. In this figure we can see that, there are 6 categorical values which are good, moderate, unhealthy for sensitive subjects, unhealthy, very unhealthy, and hazardous. Each of the categorical values is defined with different colors and declares the health implications and cautionary statement for PM2.5.

The colors can refer to a certain level of pollutant PM2.5 particle which is present in the air and this live data reading can be taken from [28]. By reading these data through image processing AI algorithms can say which area is how polluted and who will be affected by these pollutions.

Figure 3.3 shows us a reading of live data through a live server which indicates the PM2.5 particle pollution.

As in the above picture, we can see that the reading from all over the world can be taken from such kind of database. However, it is very difficult to take such a big amount of data and bring out the minimal result. As a result, from taking the help of the Dhaka US Consulate Air Pollution: Real-time Air Quality Index [28], we find

¹[\\$Dhaka](https://www.airnow.gov/international/us-embassies-and-consulates/#Bangladesh)



Figure 3.3: Air quality image worldwide [28]

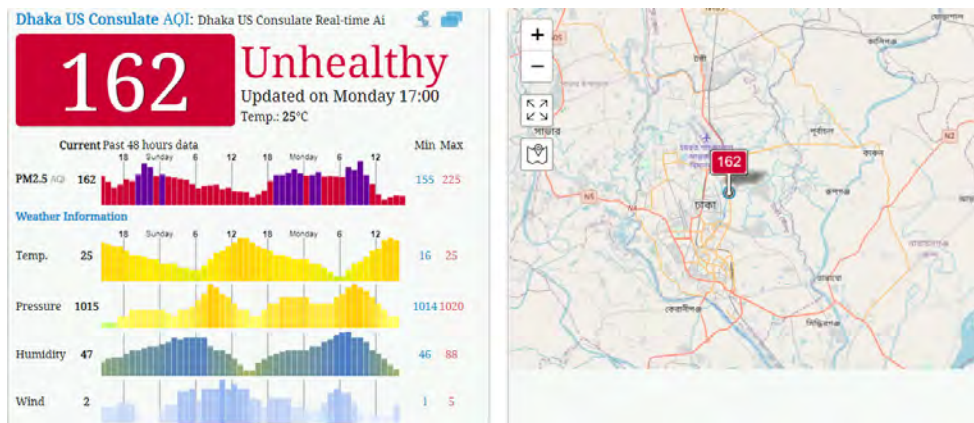


Figure 3.4: Air quality and PM2.5 pollution in Dhaka [28]

AQI	Air Pollution Level	Health Implications	Cautionary Statement (for PM2.5)
0 - 50	Good	Air quality is considered satisfactory, and air pollution poses little or no risk	None
51 -100	Moderate	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
101-150	Unhealthy for Sensitive Groups	Members of sensitive groups may experience health effects. The general public is not likely to be affected.	Active children and adults, and people with respiratory disease, such as asthma, should limit prolonged outdoor exertion.
151-200	Unhealthy	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects	Active children and adults, and people with respiratory disease, such as asthma, should avoid prolonged outdoor exertion; everyone else, especially children, should limit prolonged outdoor exertion
201-300	Very Unhealthy	Health warnings of emergency conditions. The entire population is more likely to be affected.	Active children and adults, and people with respiratory disease, such as asthma, should avoid all outdoor exertion; everyone else, especially children, should limit outdoor exertion.
300+	Hazardous	Health alert: everyone may experience more serious health effects	Everyone should avoid all outdoor exertion

Figure 3.5: AQI Category and Health Implications [9]

that the air quality in Dhaka is very unhealthy which scores 162 to date. Figure 3.4 shows the visual representation of this situation.

While these values are categorical and needed to be determined as numerical values before pre-processing, we declared Good as 0, Hazardous as 1, Moderate as 2, Unhealthy for the sensitive groups as 3, Unhealthy as 4, and Very unhealthy as 5.

3.1.1 PM2.5 detection Device

To detect PM2.5 particles in the air, first we have to build an air quality sensor with Arduino. With the help of a pretty much straightforward arduino code, we can measure the particle concentration for dust in the air. It will reveal the size and quantity for PM2.5. To build the device, first we used a sensor named PMS5003. PMS5003 is a digital particle concentration sensor which has been used worldwide. This device can acquire the exact number of ceased particles which lie in the environment. Particulate matter having a particle diameter of up to 2.5 microns or 10 microns, referred to as PM2.5 and PM10, is one of the most hazardous contaminants in the atmosphere. In addition to provoking asthma episodes and contributing to cardiovascular illness, PM2.5 particles may go deep into lungs because of their tiny size. There is a danger in having high levels of dust or PM in the air. PM2.5 has a diameter of less than 2.5 microns, while PM10 has a diameter of less than 10 microns. In other words, a PM10 report includes a report on PM2.5 as well. A human hair has a diameter of roughly 70 microns, therefore none of these particles are even close. All types of combustion in the air like motor mediums, fervent power plants, residential fuel flaming, wood conflagration, agronomic burning, and other manufactured procedures generate PM2.5 particles. The mean P.M2.5 margin for 24 hour is 35 μ g/m³.

We purchased these subsequent components to build the device. The components are given below:

Serial Number	Components	Description	Quantity
1	Arduino Board	Arduino Uno R3 Development board	1
2	P.M2.5 Sensor	PMS5003	1
3	LCD Display	JHD204A 20X4 LCD Display	1
4	Potentiometer	10K	1
5	Connecting wire	Jumper wires	15
6	Breadboard	-	1

Table 3.1: Table of the components description

3.1.2 PMS5003 Air Quality Sensor

The laser particle counter from Plantower, along with the PMS1003, PMS3003, and PMS7003, is a low-cost alternative to more costly Plantower sensors. As a digital

and universal particle concentration sensor, the PMS5003 is more than efficient at obtaining and displaying airborne particle concentrations in an easy-to-read digital format. As a result of this sensor's versatility, it may be used to monitor the intentness of ceased particles in the air and other environmental reformation equipment. A laser dispersing concept is used to create scattering, which is then collected to get the scattering light's change in time curve. Laser scattering generates scattering by radiating airborne particles with a laser. In the end, microprocessor-based MIE theory may be used to ordain the corresponding droplet size and the amount of molecules per unit volume.

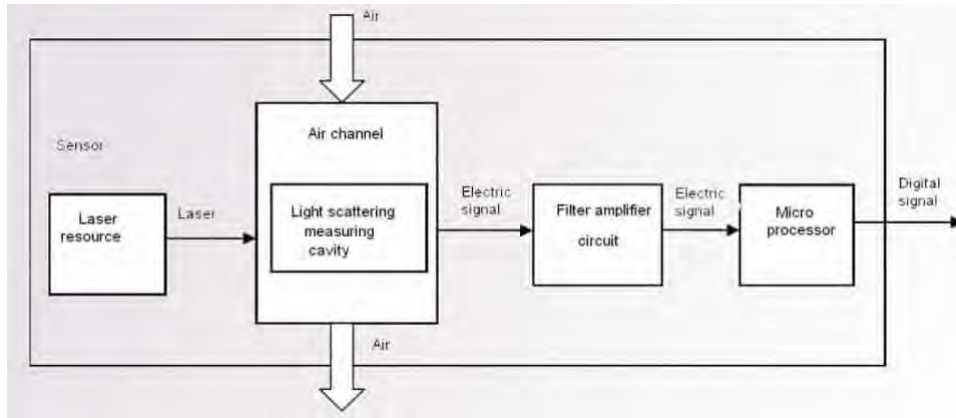


Figure 3.6: AI for Clean Air [22].

We should count the PM2.5 pins from left side to right side just as 1, 2, 3,8. But in the case of PM5003, it's different because the counting in this case of pins is designated from right to left. We were careful when we were connecting PM5003 pins as there is a chance to connect it reversely.

Pin	Description	Comments
1	Supply voltage - (VCC)	4.5-5.5V
2	Ground - (GND)	
3	HIGH or SUSPENDED- work LOW-sleep mode -(SET)	3.3V logic
4	UART/TTL data receive - (RXD)	3.3V logic
5	UART/TTL data transmit - (TXD)	3.3V logic
6	LOW to reset - (Reset)	3.3V logic
7	Not connected - (NC)	
8	Not connected - (NC)	

Table 3.2: Description of functions.

In the original sensor, there is a connector which is used to connect the sensor with the breadboard. But in our sensor which is a chinese version, we simply cut the connector and solder the solid wire. That's why we could plant it easily.

3.1.3 Interfacing PMS5003 PM2.5 Air Quality Sensor with Arduino

It's pretty easy to connect the PMS5003 to the Arduino. We needed four connections. Initially, we connected the VCC and GND pins of the PMS5003 to the Arduino's VCC and GND pins, respectively. As displayed in the figure below, the UART pins (PIN4 Rx and PIN5 Tx) are linked to Arduino pins 3 and 4.

The serial data sent by the Plantower sensors at 9600 baud is readable by several computers. The sensor is connected to a computer through this USB 2.0 to TTL UART Serial Converter CP2102.

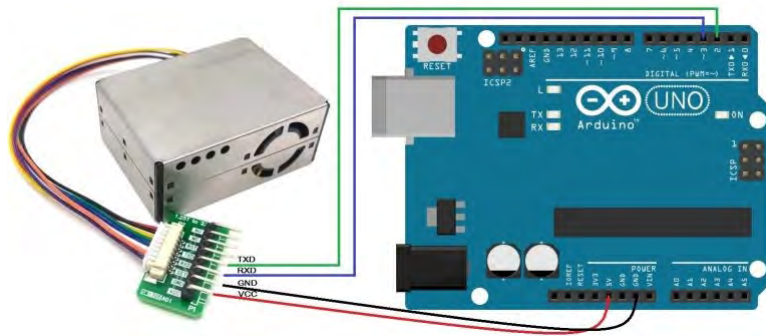


Figure 3.7: PM2.5 AQ sensor connected with Arduino [22].



Figure 3.8: Real-time PM2.5 AQ sensor connected with Arduino [22].

3.1.4 Source Code

The source code for interfacing PM2.5 PMS5003 with Arduino is given below. We just needed to upload the code to the Arduino Uno board.

Once we have uploaded the code, we opened the serial monitor, and then we set the baud rate to 9600. Then the sensor will collect the data and it will start giving the correct value after a few seconds since the sensor needs time to get warm.

```

sketch_apr19a.ino
1  #include <SoftwareSerial.h>
2  SoftwareSerial pmsSerial(2, 3);
3
4  void setup() {
5      // our debugging output
6      Serial.begin(115200);
7
8      // sensor baud rate is 9600
9      pmsSerial.begin(9600);
10 }
11
12 struct pms5003data {
13     uint16_t framelen;
14     uint16_t pm10_standard, pm25_standard, pm100_standard;
15     uint16_t pm10_env, pm25_env, pm100_env;
16     uint16_t particles_03um, particles_05um, particles_10um, particles_25um, particles_50um, particles_100um;
17     uint16_t unused;
18     uint16_t checksum;
19 };
20
21 struct pms5003data data;
22
23 void loop() {
24     if (readPMSdata(&pmsSerial)) {
25         // reading data was successful!
26         Serial.println();
27         Serial.print(data.pm25_standard);
28         Serial.print("\t");Serial.print(data.pm25_env);
29     }
30 }
31
32 boolean readPMSdata(Stream *s) {
33     if (! s->available()) {
34         return false;
35     }
36
37     // Read a byte at a time until we get to the special '0x42' start-byte
38     if (s->peek() != 0x42) {
39         s->read();
40         return false;
41     }
42
43     // Now read all 32 bytes
44     if (s->available() < 32) {
45         return false;
46     }
47
48     uint8_t buffer[32];
49     uint16_t sum = 0;
50     s->readBytes(buffer, 32);
51
52     // get checksum ready
53     for (uint8_t i=0; i<30; i++) {
54         sum += buffer[i];
55     }
56
57     // The data comes in endian'd, this solves it so it works on all platforms
58     uint16_t buffer_u16[15];
59     for (uint8_t i=0; i<15; i++) {
60         buffer_u16[i] = buffer[2 + i*2 + 1];
61         buffer_u16[i] += (buffer[2 + i*2] << 8);
62     }
63
64     memcpy((void *)&data, (void *)buffer_u16, 30);
65
66     if (sum != data.checksum) {
67         Serial.println("Checksum failure");
68         return false;
69     }
70     return true;
71 }

```

Figure 3.9: Source code

3.1.5 Interfacing PMS5003 PM2.5 Air Quality Sensor with Arduino LCD Display

To interface PMS5003 with Arduino and an LCD, four connections were required. We connect the PMS5003's PIN1 VCC and PIN2 GND to the Arduino's 5V and GND pins, respectively. The UART pins (PIN4 Rx and PIN5 Tx) are linked to Arduino pins 3 and 4. Connect pins 1, 3, and 16 to GND and pins 2 and 15 to VCC 5V for the LCD 20x4. Now connect the pins 4, 6, 11, 12, 13, and 14 of the LCD serially to the pins 13, 12, 11, 10, 9, 8, and 8 of the Arduino. Then, to adjust the contrast, we also connected a 10K potentiometer to LCD pin 3.

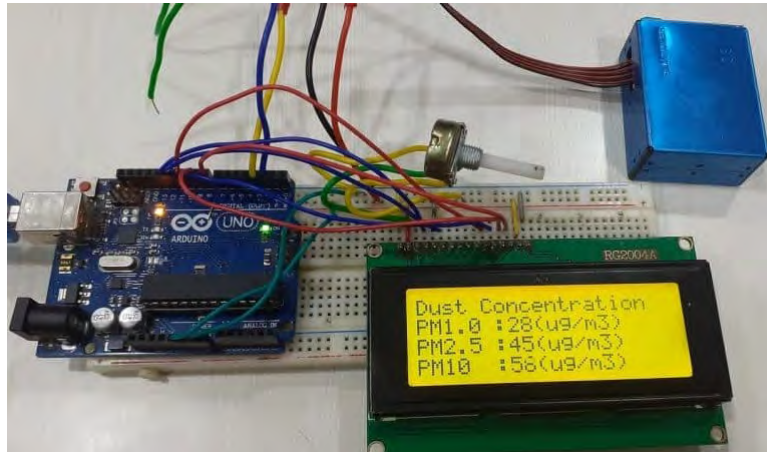


Figure 3.10: Interfacing PMS5003 PM2.5 with Arduino LCD Display

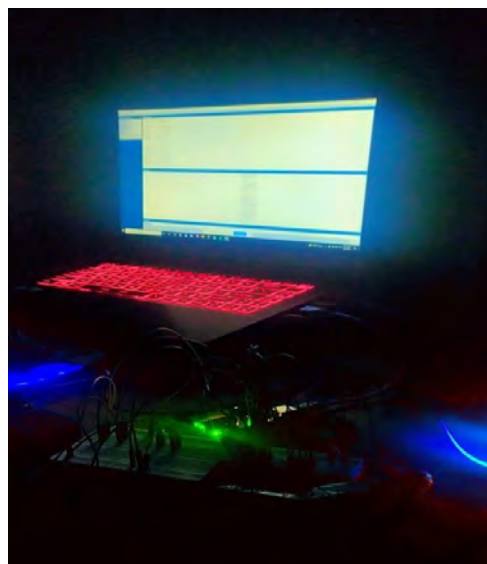


Figure 3.11: We are getting data using the device

3.1.6 Source Code for LCD

We have attached the source code for interfacing the sensor we used which is PMS5003 and LCD display. We simply copied the code and uploaded it to the

Arduino UNO board. But first, we wanted to add the library for PMS5003 sensor. To download it we simply need to install the PMS Library by Mariusz Kacki Version 1.1.0 from the library manager.

```
sketch_may17a.ino
1  #include "PMS.h"
2  #include "SoftwareSerial.h"
3  #include <LiquidCrystal.h>
4  LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
5
6  SoftwareSerial Serial1(2, 3); // RX, TX
7
8  PMS pms(Serial1);
9  PMS::DATA data;
10
11 void setup()
12 {
13     Serial1.begin(9600);
14     lcd.begin(20,4);
15     lcd.setCursor(0, 0);
16     lcd.print("warming up");
17     delay(4000);
18     lcd.clear();
19 }
20
21 void loop()
22 {
23     if (pms.read(data))
24     {
25         lcd.clear();
26         lcd.setCursor(0, 0);
27         lcd.print("Dust Concentration");
28         lcd.setCursor(0, 1);
29         lcd.print("PM1.0 : " + String(data.PM_AE_UG_1_0) + "(ug/m3)");
30         lcd.setCursor(0, 2);
31         lcd.print("PM2.5 : " + String(data.PM_AE_UG_2_5) + "(ug/m3)");
32         lcd.setCursor(0, 3);
33         lcd.print("PM10 : " + String(data.PM_AE_UG_10_0) + "(ug/m3)");
34
35         delay(1000);
36     }
37 }
```

Figure 3.12: LCD source code

3.1.7 Storing Data

After uploading the code to Arduino, the sensor will begin collecting data as it receives power. The power supply can be given through a working device like Laptop or any other external power source. When the sensor begins to gather data, the data will be shown on the PC's serial monitor or the Arduino's built-in serial monitor. We can also display the data on the LCD display by applying the LCD display's source code. Now we must save the data in order to incorporate it into our algorithm. For this, we have several options, including Microsoft's Excel streamer and Arduino's Arduspreadsheet. We used an Arduspreadsheet to collect our data. Generally, Arduspreadsheets are not available in Arduino. For this purpose, we have downloaded Indrek Luuk's Arduspreadsheet plug-in for the Arduino IDE to export serial data as a CSV file that can be opened in any spreadsheet program. The plug-in file must be placed in the "tools" folder adjacent to the "Libraries" folder in Arduino. Following a restart of Arduino IDE, "Arduspreadsheet" should be accessible through the "tools" menu. After accessing Arduspreadsheet, the "start" option must be selected. After that, we should notice the data being collected on a

spreadsheet. Once the necessary quantity of data has been obtained, it can be saved as a CSV file. By clicking the "add timesteps" box, we can also add timesteps.

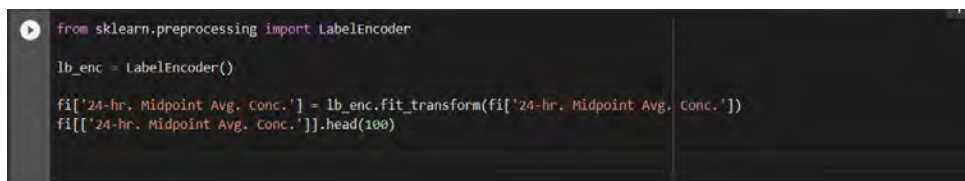
3.2 Data Pre-processing

When it comes to the fact of data pre-processing, we need to consider many things. The most difficult challenge, in this case, was to process a numerous amount of data worldwide. As a result, the data was centralized to a certain area, and rather than concentrating on the amount of data we have taken the least amount of data from a certain area to keep it more authentic. Therefore, we have taken data from Dhaka US Consulate Air Pollution: Real-time Air Quality Index [28] and put it on a Microsoft Excel file named "Dhaka-AQI-MTD.cvs". There are about 4080 rows and 14 columns of data which was processed. To add more, unnecessary attributes were deleted because the proposed model only needs the AQI of PM2.5 and AQI category and the 24-hour concentration of PM2.5 on the air quality.

Also, the data set was arranged in a lowest to the highest order because this representation of the organization was followed. After all, it helps to label the data properly which will help to find out the air quality index manually, and also if there is any data that is giving any error in the code snippet can also be taken care of.

Another thing to mention, while pre-processing data we need to convert categorical values into numerical values. The machine cannot read categorical values rather it has been translated as a numerical values. As we have used python here to pre-process the data, we firstly needed we converted categorical values into a numerical values and then did the pre-processing Figure 3.12 shows the pre-processing algorithm that is used for converting categorical values into numerical values.

Notably, as our data-set has 14 columns and also some negative values including 0 the processed data will not have all of them. Some of the columns include parameter, date, year, day and hour will be dropped only AQI of PM2.5 and AQI category and the 24-hour concentration of PM2.5 on the air quality will be shown of 4080 records. The value of each attribute is real and can be represented in figure 3.13.



```
from sklearn.preprocessing import LabelEncoder

lb_enc = LabelEncoder()

fi['24-hr. Midpoint Avg. Conc.'] = lb_enc.fit_transform(fi['24-hr. Midpoint Avg. Conc.'])
fi[['24-hr. Midpoint Avg. Conc.']].head(100)
```

Figure 3.13: Categorical Value Conversion

data

	Site	Parameter	Date (LT)	Year	Month	Day	Hour	AQI	AQI Category	24-hr. Midpoint Avg. Conc.	Raw Conc.	Conc. Unit	Duration	QC Name
0	Dhaka	PM2.5 - Prindpal	12/1/2017 1:00	2017	12	1	1	204.2	254	Very Unhealthy	217	UG/M3	1 Hr	Valid
1	Dhaka	PM2.5 - Prindpal	12/1/2017 2:00	2017	12	1	2	189.6	240	Very Unhealthy	175	UG/M3	1 Hr	Valid
2	Dhaka	PM2.5 - Prindpal	12/1/2017 3:00	2017	12	1	3	177.5	228	Very Unhealthy	166	UG/M3	1 Hr	Valid
3	Dhaka	PM2.5 - Prindpal	12/1/2017 4:00	2017	12	1	4	173.9	224	Very Unhealthy	170	UG/M3	1 Hr	Valid
4	Dhaka	PM2.5 - Prindpal	12/1/2017 5:00	2017	12	1	5	172.9	223	Very Unhealthy	169	UG/M3	1 Hr	Valid
...
4075	Dhaka	PM2.5 - Prindpal	12/31/2021 21:00	2021	12	31	21	106.7	177	Unhealthy	116	UG/M3	1 Hr	Valid
4076	Dhaka	PM2.5 - Prindpal	12/31/2021 22:00	2021	12	31	22	115.8	182	Unhealthy	125	UG/M3	1 Hr	Valid
4077	Dhaka	PM2.5 - Prindpal	12/31/2021 23:00	2021	12	31	23	125.4	187	Unhealthy	135	UG/M3	1 Hr	Valid
4078	Dhaka	PM2.5 - Prindpal	1/1/2022 0:00	2022	1	1	0	127.7	188	Unhealthy	130	UG/M3	1 Hr	Valid
4079	Dhaka	PM2.5 - Prindpal	1/1/2022 1:00	2022	1	1	1	131.3	190	Unhealthy	135	UG/M3	1 Hr	Valid

4080 rows x 14 columns

[Before Pre-processing]

	AQI	AQI Category	24-hr. Midpoint Avg. Conc.
0	204.2	254	5
1	189.6	240	5
2	177.8	228	5
3	173.9	224	5
4	172.9	223	5
...
4075	106.7	177	3
4076	115.8	182	3
4077	125.4	187	3
4078	127.7	188	3
4079	131.3	190	3

[After Pre-processing]

Figure 3.14: Input data before and after pre-processing

Chapter 4

Implementations and Result

This section describes the implementation of the proposed module for AI-assisted Framework for Reducing Air Pollution with The Help of High-Quality Image Processing and Computational Cloud Computing. The model was implemented and tested using python in google colab. The model consists of two stages which are Data pre-processing and applying the algorithm. Input data pre-processing is a stage for qualifying input data for the detection of air pollution. Logistic regression and Decision tree are used for detecting the accuracy in terms of reading the AQI from the data set.

4.1 Implementation

4.1.1 Logistic Regression

The method of modeling the probability of a discrete result given an input variable is known as logistic regression. In machine learning, it is used for perfect predictions. For classification situations, where we want to see if a new sample fits well into a category, logistic regression is a good analytical tool. Moreover, when there are several explanatory variables, logistic regression is used to calculate the odds ratio. With the exception that the response variable is binomial, the approach is quite similar to multiple linear regression. The influence of each variable on the odds ratio of the observed event of interest is the end outcome. The key benefit of logistic regression is that it eliminates confusing effects by examining the relationship between all variables.

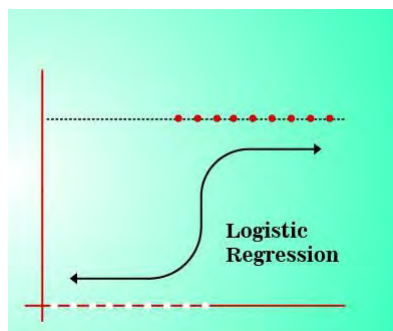


Figure 4.1: Logistic Regression [2]

Here, in figure 4.2 is the code of the accuracy score that we used for Logistic Regression and our accuracy was 0.9963.

```
▶ from sklearn.linear_model import LogisticRegression
  modl = LogisticRegression()
  modl.fit(X_train, y_train.values.ravel())
  prdc = modl.predict(X_test)
  from sklearn.metrics import accuracy_score
  LinRegSco=accuracy_score(y_test, prdc)
  print(LinRegSco)
↳ 0.9963235294117647
```

Figure 4.2: Logistic Regression Accuracy Test.

4.1.2 Decision Tree

A Decision Tree is a supervised learning technique that may be used to solve both classification and regression problems, however, it is most commonly used to solve classification issues. Internal nodes represent dataset attributes, branches represent decision rules, and each leaf node provides the conclusion in this tree-structured classifier. It's a graphical representation for finding all effective solutions to a problem/decision depending on certain parameters. Decision Tree algorithms are designed to simulate human thinking abilities when making decisions, making them simple to comprehend. Because the decision tree has a tree-like form, the rationale behind it is simple to comprehend. It helps to think about all the possible outcomes for a problem.

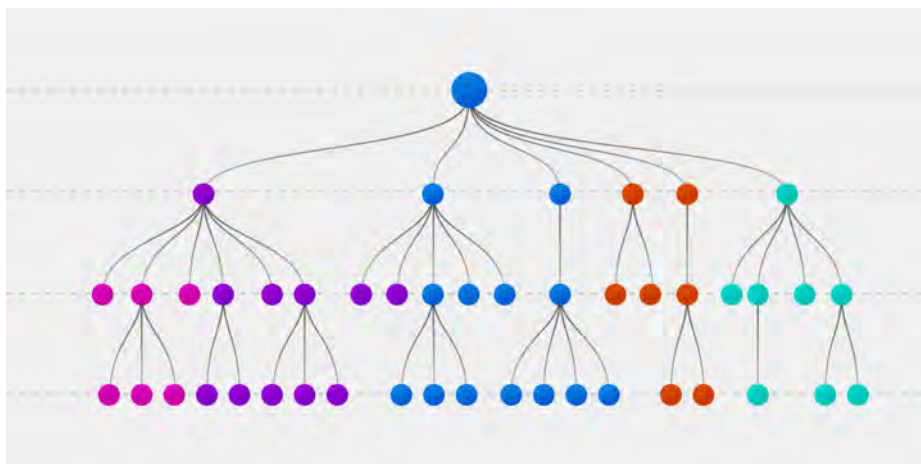


Figure 4.3: Decision Tree [7]

Here, in figure 4.4 is the code of the accuracy score that we used for the Decision Tree and our accuracy was 0.9975.

```
▶ from sklearn.tree import DecisionTreeClassifier
  clfdt = DecisionTreeClassifier(criterion='entropy',random_state=1)
  clfdt.fit(X_train,y_train)
  y_prd = clfdt.predict(X_test)
  dtscor=accuracy_score(y_prd,y_test)
  print(dtscor)

↳ 0.9975490196078431
```

Figure 4.4: Decision Tree Accuracy Test.

4.1.3 GaussianNB Algorithm

The full abbreviation of the GaussianNB algorithm is Gaussian Naive Bayes algorithm. It is one kind or a representative of Naïve Bayes algorithm. This algorithm represents and at the same time models features consisting continuous values fitting to a Gaussian range or allocation. Basically, the features are supposed to be following the Gaussian or the normal distribution. There is a group of supervised machine learning algorithms, consisting many algorithms which are used in machine learning. This group or family or collection is basically known as Naïve Bayes. The Bayes theorem also comes under this family as well as it supports the basic criteria. This particular algorithm follows a technique to perform classification which has a lot of potential. Especially when the task consists inputs which have a number of dimensions. Adding to that it can prove to be handier when the task is to solve classification issues which are too complex. There are Bayes theorem, naive Bayes classifier and Gaussian Naïve Bayes. Firstly, the conditional probability may be calculated using Bayes' Theorem. It is used in Machine Learning because it is a valuable technique for the study of probability. Secondly, The Bayes Theorem is used to create Naive Bayes Classifiers. The high freedom assumptions between the characteristics are one of the assumptions made. This basically means the classifiers created by the Bayes Theorem make an assumption that each feature has its own importance. It doesn't really depend on any other characteristics. So whatever worth other characteristics hold, it doesn't matter to the particular feature's importance. These classifiers, meaning the Naïve Bayes Classifiers are very effective. Specifically in supervised learning situations. The classifier uses training data to figure out variables to be used for classification. But here the Naïve Bayes Classifier doesn't require a whole lot of them. It can work with a very minimum number of data. Another effectiveness of this classifier can be described with the easiness of development and execution of them. Finally, as the algorithm works with continuous data and while doing so it's aware that the continuous values which are linked with each class do follow a normal distribution. So as touched upon a bit earlier, the Gaussian Naive Bayes accepts continuous numerical features and models them all as Gaussian (normal) distributions. To build a basic model, assume the data is characterized by a Gaussian distribution with no covariance (independent dimensions) between the parameters. This model may be fitted by simply calculating the mean and standard deviation of the points within each label, which is all that is required to construct a distribution of this type. [18].

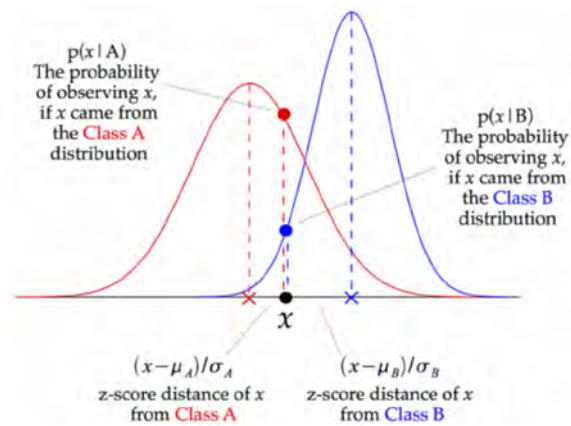


Figure 4.5: Gaussian Naïve Bayes Algorithm [18]

The z-score distance between each data point and each class mean is determined. As a result, we can observe that the Gaussian Naive Bayes technique is slightly different and can be employed well.

Here, in figure 4.6 is the code of the accuracy score that we used for the GaussianNB algorithm and our accuracy was 99.6323.

```

from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train.values.ravel())
y_pred_gaus = gnb.predict(X_test)
gaus_score = accuracy_score(y_pred_gaus, y_test) * 100
print(gaus_score)

99.63235294117648

```

Figure 4.6: Gaussian Naïve Bayes Algorithm Accuracy Test.

4.1.4 Gradient Boosting Algorithm

To describe the Gradient Boosting algorithm properly we can say that it's a method which literally uses learnings from all of the weak learners to eventually build a strong model. So basically, it can be called a boosting method. The Gradient Boosting is mainly a useful algorithm, especially in terms of solving tasks related to regression as well as tasks from the topic of classification. As mentioned above the algorithm takes weak prediction models. These are mostly decision trees. Then by putting them together it comes up with a strong prediction model. In this case all the weak models after being turned in to a stronger model by an ensemble, helps in reduction of the average of squared error of the whole model. Which is basically the mean squared error or MSE. Also, this algorithm is very useful as after the stronger model is built from taking learnings from the weak models, it starts to provide a value of accuracy which is unbeatable. This is the predictive accuracy we are talking about. This model isn't constructed in any different style than the other approaches serving the same purpose. That is in the similar stage-wise manner. But in one particular case it differs from those other methods out there. That is with this model any

differentiable loss function can be optimized. In the realm of machine learning, the gradient boosting method is one of the most powerful. Bias and Variance errors are two types of faults in machine learning systems. As one of the boosting strategies, gradient boosting is used to reduce the model's bias error. Moreover, the gradient boosting approach may be used to forecast not just continuous but also categorical target variables (as a Regressor) (as a Classifier). Mean Square Error is the cost function when used as a regressor, while Log loss is the cost function when used as a classifier [21].

Gradient boosting has three components:

- 1) A loss function that has to be improved.
- 2) To make forecasts, a poor learner.
- 3) To reduce the loss function, an additive model is used to incorporate weak learners.

So, this particular algorithm of Gradient Boosting goes deeper and stronger at the same time.

Here, in figure 4.7 is the code of the accuracy score that we used for the Gradient Boosting algorithm and our accuracy was 98.6519.

```
from sklearn.ensemble import GradientBoostingClassifier
gbc = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0)
gbc.fit(X_train, y_train.values.ravel())
y_pred_gbc = gbc.predict(X_test)
mlp_score = accuracy_score(y_pred_gbc, y_test)*100
print(mlp_score)

98.65196078431373
```

Figure 4.7: Gradient Boosting Algorithm Accuracy Test

4.1.5 SVC Algorithm

The full meaning of SVC is Support vector clustering. SVC is a nonparametric segmentation technique that makes no assumptions about the size or structure of the data clusters. Because it works best for low-dimensional data, a preprocessing phase, such as principal component analysis, is typically necessary if your data is high-dimensional. Here the goal is to make the data more intelligible. In order to do that it divides the data collection into various clusters or groups. This division is done based on some characteristics. There are several options for reaching this aim. Clustering can be done using a parametric model or using a distance or similarity measure, as in hierarchical clustering. Cluster boundaries are naturally placed in regions of the space of data for example “valleys” where there are little data in the data's probability distribution. Furthermore, Support vector clustering (SVC), which is founded on the support vector technique, takes this route. A kernel function is used to translate SVC data points from the dataset

to a high-dimensional feature space. By using the Support Vector Domain Description approach, the kernel's feature space is searched for the minimal sphere that encloses the picture of the data. When projected back to data space, this sphere creates a collection of outlines that encircle the data points. The contours are then read as cluster borders, and SVC associates the points encompassed by each contour to the very same cluster. There are several parameters of this SVC algorithm that are very helpful to implement the algorithm some of them are *kernel_gamma*, *add_cluster_attribute*, *add_as_label*, *remove_unlabeled*, *min_pts*, and *kernel_type* [5].

Here, in figure 4.8 is the code of the accuracy score that we used for the SVC algorithm and our accuracy was 0.9963.

```

from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state = 1)
classifier.fit(X_train,y_train)
s_prdd = clf.predict(X_test)
#print(s_prdd)
svcscore=accuracy_score(s_prdd,y_test)
print(svcscore)
0.9963235294117647

```

Figure 4.8: SVC Algorithm Accuracy Test.

4.1.6 Passive Aggressive Classifier

Many advanced Machine Learning devotees, even apprentices, are unaccustomed with the Passive-Aggressive algorithms. This algorithm is a category of Machine Learning algorithms. However, they can be perfectly helpful and efficient in several conditions. For the purpose of a bigger learning purpose, this algorithm is used often. This particular algorithm is included in those kinds which are used as the so-called “Online-learning algorithms”. Yet, like most algorithms which learn in batch, where the whole training dataset is used at once, it is different in that. The algorithms termed as these, first take the data which are given as input and then update the machine learning model in a sequential sequence step by step. It is most recommended in a situation where the quantity of data is large it is impossible to train the full dataset computationally because of bulk data. An online learning algorithm most commonly obtains a training example under the system, the classifier is modified, and discards the sample. Moreover, it also detects the false data on the bulk data in a constant updating dataset while new data is generated every moment which can be a great use of this. The quantity of data required for continuous reading data from the data set can be a lengthy task. As a result, this kind of algorithm is perfect [16].

Passive-Aggressive Algorithms in Action:

The algorithms are known as passive-aggressive because:

1) If the forecast is right, maintain the model and don't modify anything; meaning, it is not possible to come up with any sort of adaptation for the model as in this


```

from sklearn.linear_model import PassiveAggressiveClassifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(X_train,y_train)
pac_prdd = pac.predict(X_test)
#print(pac_prdd)
pacscor=accuracy_score(pac_prdd,y_test)
print(pacscor)
0.9963235294117647

```

Figure 4.9: Passive Aggressive Classifier Accuracy Test.

case the data is not sufficient enough.

2) It is called aggressive because it says, if the forecast is not on target, then there should be adaptation made accordingly, to the model. Suggesting, the issue might get solved through a modification in the design.

Here, in figure 4.9 is the code of the accuracy score that we used for the Passive-Aggressive Classifier algorithm and our accuracy was 0.9963.

4.1.7 SGD Algorithm

It is a Sklearn optimization algorithm called which is known as the Stochastic Gradient Descent algorithm. Stochastic means referring to a system also, a process that has an erratic probability associated with it. SGD is more likely to be a basic and also an optimization system that is very effective in the process of determining a set of variables of functions which reduces a cost function. In other words, it is utilized for learning discriminative linear classifiers using SVM and Logistic Regression which are also known as convex loss functions. As a result, because the upgrade in the coefficients is conducted for every training sample instead of the end of every example, this algorithm has successfully been used in datasets that are large-scaled. Moreover, in this algorithm, there is a huge advantage with the classifier. As it is a method which can be called more of a typical SGD learning one. And it accepts more than one loss functions as well as at the same time more than one classification penalties. Therefore, rather than choosing the whole data set for each iteration in SGD, a small number of samples are randomly chosen. Now, the “batch” term is used while working with SGD because it indicates and calculates the gradient from each iteration from the total number of sample data in a dataset [20]. The batch is considered to be the whole dataset in standard Gradient Descent optimization, such as Batch Gradient Descent. Although considering the entire dataset is quite valuable for reaching the minima less noisily and randomly, the challenge emerges as our datasets become large. To implement SGD classification, Scikit-learn offers the SGD Classifier module. If one’s dataset has a million samples, to finish one iteration successfully, of the Gradient Descent we are talking about here, one must utilize the whole of that dataset’s samples and this must be repeated until the minima are achieved. As a result, doing it becomes computationally costly. To overcome this issue, SGD comes in as an option. In case of SGD, instead of performing iteration with a million samples, it does it with only one. That means, the batch size with what the iteration here is done with is one. While performing the iteration, a random sample gets chosen from a mix of the samples [30].

$$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta) \quad (4.1)$$

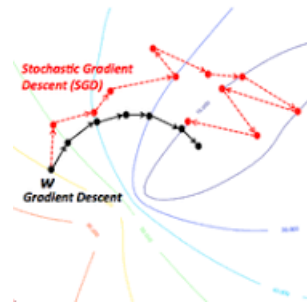


Figure 4.10: SGD Algorithm [30]

```

from sklearn.linear_model import SGDClassifier
sgd = SGDClassifier(loss="hinge", penalty="l2", max_iter=5)
sgd.fit(X_train,y_train)
sgd_prdd = sgd.predict(X_test)
#print(sgd_prdd)
sgdscor=accuracy_score(sgd_prdd,y_test)
print(sgdscor)
0.9963235294117647

```

Figure 4.11: SGD Algorithm Accuracy Test.

Here, in figure 4.11 is the code of the accuracy score that we used for the SGD algorithm and our accuracy was 0.9963.

4.1.8 MLP Classifier Algorithm

If we do a breakdown of the name of this Classifier, we find multi for ‘M’, layer for ‘L’ and perception for ‘P’. It is a classifier that has its links with neural network. MLP classifier, Algorithms like Naïve Bayes or Vector algorithms which are referred to as classification algorithm uses an underlying Neural Network to execute the classification task. A multilayer perception is mainly a neural network that is able to connect many layers of data in the directed graph which means it will keep the route of the signal in only one direction throughout the nodes. A non-linear activation function exists for every node except the input nodes. The direct learning approach is used in MLP in order to achieve back propagation. MLP is a deep learning approach because there are a great number of neurons. Also, MLP algorithms are very well trained where the input data are classified or numbered for classification and prediction issues. They may also be used to perform a prediction in quest of an output of real values after some inputs are provided in regression situations. The more regular use of MLP is found tasks related to supervised learning. Adding to that it can also be useful in the section of cognitive neuroscience and parallelism distributed processing analysis. MLP Classifier iterates because smaller parts of the loss function concerning the design variables are generated at each time step to update the parameters. A regularization factor can also be introduced to the loss function to decrease model parameters and prevent over fitting. This method works with floating-point data stored in packed NumPy arrays. Speech recognition, picture recognition, and machine translation are examples of applications. Furthermore, MLPs are useful for Tables of data, Prediction issues in classification, and Problems with regression prediction [29].

Here, in figure 4.12 is the code of the accuracy score that we used for the MLP Classifier algorithm and our accuracy was 98.6519

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(7), activation="relu", max_iter=10000)
mlp.fit(X_train, y_train.values.ravel())
y_pred_mlp = mlp.predict(X_test)
#print(y_pred_mlp)
mlp_score = accuracy_score(y_pred_mlp, y_test)*100
print(mlp_score)
98.65196078431373
```

Figure 4.12: MLP Classifier Accuracy Test.

4.1.9 KNN Algorithm

KNN is basically the short form for K-Nearest Neighbor. It is an algorithm which falls under the category of machine learning and at the same time is a supervised one. This algorithm has a wide range of usage. Mainly the algorithm works to solve tasks related to classification and regression. In the case of the input in this algorithm, it consists of the closest training examples in a data set. This is for both classification and regression. And then the output depends on which case the algorithm is being used. The outputs will vary for classification and regression. The ‘K’ in the KNN algorithm stands for a number that will be the closest neighbor to an unknown variable and which needs to be classified. So basically, this algorithm aims to take data and assess it. Then classify it according to its nature. Where the data fits the best. The KNN algorithm tries to find out the closest neighbor according to similarity around the data based on the assessment done before. So to perform the task determining the value of K is important. But a determined solution or way to get a value of K which can be termed as fully correct is non-existent. So the K here that provides the highest accuracy for both training and testing data is chosen. After determining the value of K, now the task is to perform classification. KNN prefers to apply the concept of “majority voting” when the problem statement is of the “classification” type. The class with the most votes is picked within the given range of K values. It’s more like a general election that takes place in a republic country. Several parties stand to fight in the election. But the one that gets the majority of votes wins. In this case too, given a particular range which is denoted by K, the class that is in the majority number is selected. This is how this particular algorithm works when the task is related to classification. Now there is another different part of its usage. Regression. In this case, our algorithm uses a different method. A mean based method is used to project the desired merit of data that can be labelled as fresh. While performing this method, the algorithm takes the neighbors that are the closest into consideration. Here this task is done with regard to K. It basically collects the values from those selected nearest neighbors and performs a calculation of the mean. This process is run until all those values are found inside a certain range. The value of K defines the range here. Now the algorithm creates a problem when the data set is imbalanced. If the data in the data set isn’t balanced properly then the KNN algorithm gets biased. It gives biased results. So to avoid that the data set should be balanced. This balancing can be done by the up-scaling or down-scaling method. So this is how the KNN algorithm works. In short, it’s an algorithm used for classification or regression which works basically

with a distance-based approach [25].

Here, in figure 4.13 is the code of the accuracy score that we used for the KNeighbors Classifier algorithm and our accuracy was 99.6323.

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(X_train, y_train.values.ravel())
y_pred_knn = knn.predict(X_test)
#print(y_pred_knn)
dts_score=accuracy_score(y_pred_knn,y_test)*100
print(dts_score)

99.63235294117648
```

Figure 4.13: K Neighbors Classifier Algorithm Accuracy Test

4.1.10 Linear SVC

A Linear SVC is an algorithm that should be fitting to the data that has been provided. It should also at the same time fulfill the requirement of providing a hyper-plane that will suit best which will come into use for the data to be categorized. After obtaining the hyper-plane, some characteristics might be put as input to the classifier to get the predicted class. That's how a linear SVC functions. The main goal is to perform classification. The Linear SVC algorithm performs classification and the benefit of using this algorithm is it can work very well when the scenario provides data at a big scale. If we put the models of SVC and the Linear SVC side by side to run a differentiation between them, it will be found that the later has some extra features to it which makes it preferable. The Linear SVC works with some additional parameters. Penalty normalization is one of them which applies loss function. To describe the process in short of how classification is done by Linear SVC, we can talk about some steps. Firstly, generating a classification dataset that can be random or anything. Then the data in the data set need to be divided into train and test sections. Then the classifier gets defined by using the Linear SVC class. In this case, the parameters of the class can be kept to default. Or it can also be changed as the data of the classification set. The next step is to test the model using train data and check how accurate the result comes. Additionally, a cross-validation training process can also be tested to check the score of training. Finally, using the training model, the test data can be now predicted. Then to cross-check the accuracy level of the prediction the confusion matrix function can come into help. Furthermore, a classification report can also be created after this on the data that has been predicted in case we want to check the other accuracy metrics. So that's the process of how Linear SVC works to perform tasks related to classification. This particular algorithm is suited for many different situations. The Linear SVC algorithm thus can be used on various projects for suitable outcomes [15].

Here, in figure 4.14 is the code of the accuracy score that we used for the Linear SVC algorithm and our accuracy was 0.9963.

```

from sklearn.svm import LinearSVC
clf = LinearSVC(random_state=0, tol=1e-5)
clf.fit(X_train, y_train)
y_prdd = clf.predict(X_test)
#print(y_prdd)
ddtscore=accuracy_score(y_prdd,y_test)
print(ddtscore)
0.9963235294117647

```

Figure 4.14: Linear SVC Algorithm Accuracy Test.

4.1.11 Random Forest Algorithm

Random Forest is one widely used machine learning algorithm. The algorithm belongs to the supervised learning technique. It is a method that mainly works around decision trees. The algorithm is appropriate for coming up with predictions. At the same time the study of behavior is also another feature that it provides. So basically, there are decision trees. To create a picture, in a huge number. So, every piece of these decision trees mirrors a grouping or in simpler words classification, of the data that has been thrown into the random forest. These grouping for each of the decision tree is distinctive. So, what this method of random forest does is, it takes all of these affairs into account to look into them separately. And to come up with a result of this, it goes with that particular one which shows the majority number votes. Basically, it gets finalized as the selected forecast. This algorithm is best used for predictions. It can be used for both classification and regression as well. The random forest algorithm works mainly based on the idea of ensemble learning. This means adding multiple classifiers together to come up with a solution for a particular problem and at the same time improve the performance of the model. The random forest algorithm is a popular one because, among all the classification methods, this particular algorithm provides the highest accuracy. Cause to increase the accuracy of the prediction of a specific dataset, the random forest algorithm uses several decision trees on multiple subjects of that provided dataset [19]. So, it just doesn't depend on the result of one decision tree. But it evaluates gotten result from all the trees. Then depending on the bigger count of votes, that one prediction has got, it provides the final prediction of the output. That's how it increases its accuracy. So, the more the number of decision trees, the more accurate the prediction gets [22].

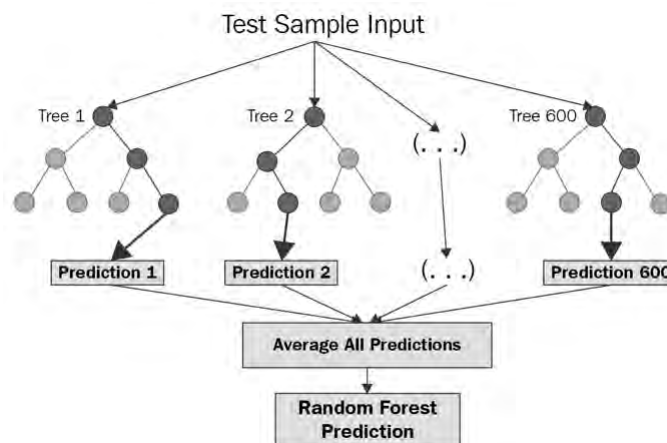


Figure 4.15: Random Forest Classifier Algorithm [19]

There are two phases of working process in this particular algorithm. Describing the first one, the random forest is created by merging several decision trees. The second phase is about making the prediction. In this phase, prediction is made for each of the trees that were created and merged in the first phase. So, describing the two phases would be like, first from the training set random K data point is selected. Then from those selected data points from the dataset, decision trees are built. These decision trees are linked with the points that were selected first from the data set. Then the next step is to choose a number N for the decision trees that need to be created or will be created. Then again it's about repeating the first two steps of selecting data points and creating decision trees. Then finally the process is to find predictions for each decision tree for the selected new data points. Then as mentioned earlier from those predictions it finds the one with the majority of votes. Then it assigns the new data points to them. That's how the random forest algorithm works basically. Another thing about the random forest algorithm is that it relies on variables since it's difficult to comprehend the models. The naive technique demonstrates the relevance of variables by attributing value to a variable based on the frequency with which all trees include it in the sample. It is simple to implement, but it poses a hurdle because the cost savings and increased accuracy are redundant. The permutation significance is a metric for tracking prediction accuracy when variables are permuted at random from out-of-bag data. The permutation importance technique outperforms the naive strategy, although it is more costly. The strategy depends on the naive, mean decrease impurity, and permutation significance techniques to provide them direct interpretability to the issues due to the challenges of the random forest not being able to interpret predictions well enough from a biological standpoint [19]. The predictor variables with numerous categories are supported by all three techniques. There are several advantages of using the random forest algorithm. One is that this algorithm can work with large datasets which gives a fair bit of flexibility. Also, it can comfortably work with datasets with high dimensionality. It gives a very accurate prediction. That's what makes this algorithm a very popular one. Because of the accuracy in the prediction the random forest algorithm makes. It also can avoid the problem of any issue with over-lifting. The algorithm can balance datasets on its own. Which can be a problem if not done properly. Balancing the dataset gets very important. In the case when any particular class is rarer than the other classes in the dataset, the random forest algorithm can balance this out. Furthermore, the variables get worked out very fast in this algorithm. This comes very handy while working with complicated tasks. In addition to all that, the random forest algorithm comes up with a solution in case of working with missing data as well. To solve the problem of missing values, the values that are missing get filled with variables that appear the most in a particular node. So the random forest algorithm comes up with so many advantages in addition to the most important one of providing the highest accuracy in terms of making predictions.

Here, in figure 4.16 is the code of the accuracy score that we used for the Linear SVC algorithm and our accuracy was 0.9963.

```

from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=50)
rfc.fit(X_train, y_train.values.ravel())
rfc_score=accuracy_score(y_prd_rf,y_test)
print([rfc_score])

0.9963235294117647

```

Figure 4.16: Random Forest Classifier Algorithm Accuracy Test.

4.2 Confusion Matrix

In this figure 4.17 all the confusion matrix of all the algorithms that we have used are shown.

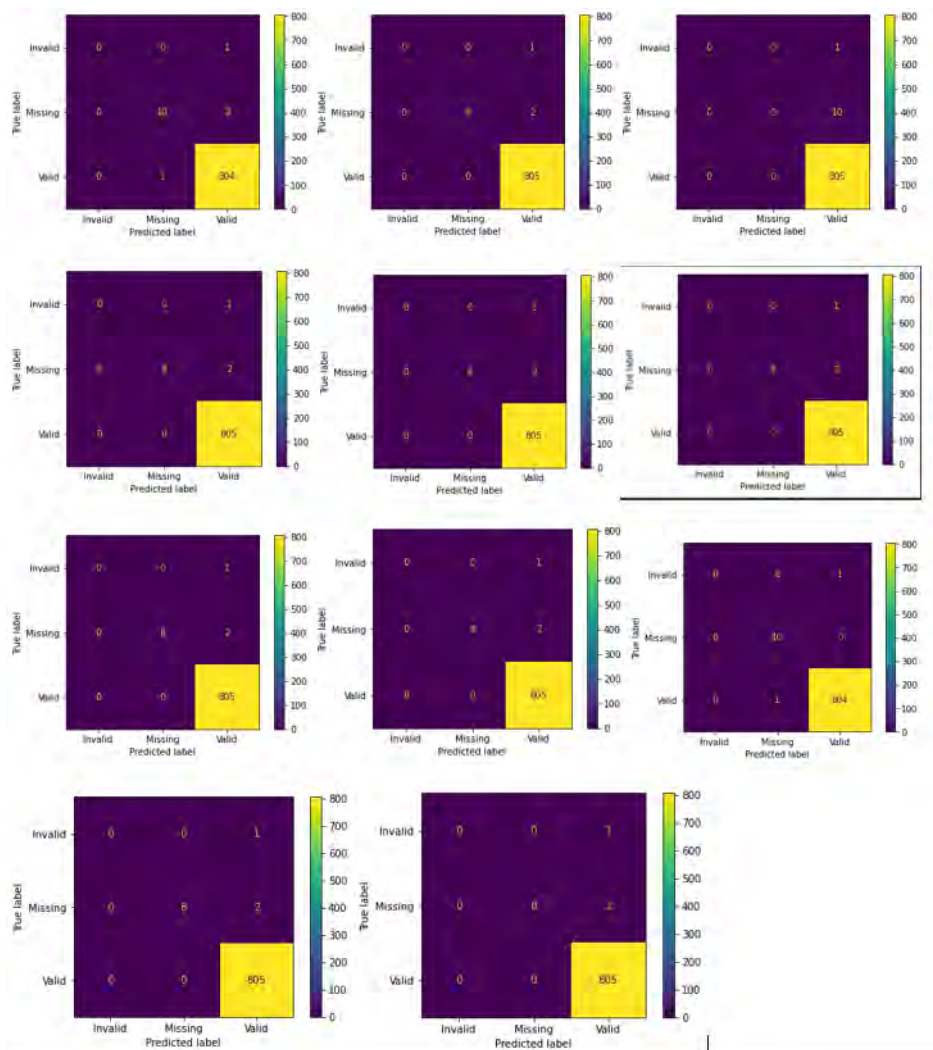


Figure 4.17: Confusion Matrix

4.3 Prediction

4.3.1 Linear Regression

Linear regression is a technique for predicting the value of one variable on the basis of the value of another variable. This technique can be termed as a statistical one. Here, the dependent variable is the variable that one wishes to forecast. On the other hand, the independent variable is the value one uses to forecast the value of the other variable. It is a very famous and recognized technique. Especially if we talk about statistics as well as machine learning. It is a very widely used practice. The predictive analysis of linear regression is a pretty basic one, which probably makes it a commonly used one as well. As mentioned above, it predicts the value of one variable, which is the outcome or dependent variable, based on the value of one or more other independent variables. To do this task, the linear regression method basically runs query on two particular things. First, it examines whether a set of variables, which can be termed as predictor variables, can properly help in predicting the outcome or dependent variable. And secondly, it runs a more specific search to detect which variables are more significant in particular to do the task. Meaning which particular variables will have more impact in predicting the value of outcome variable. At the same time the method also examines in which way these variables will be impactful in predicting the dependent variable. The magnitude and the sign of these variable's estimates will be the matter of examination in this case. There can be multiple use of linear regression. To be more precise, three. Apart from predicting values, linear regression also comes in handy in case of calculating the strength of the predictors as well as predicting the changes. Firstly, obviously the independent variables do have an effect on the dependent variables. The linear regression comes in use to determine the power of that effect. Then after that, we can understand the change that takes place in a dependent variable with the help of linear regression. Surely the idea here is to use one or multiple variables' values to predict the value of a variable. So, a change does take place in the outcome variable. The analysis of linear regression helps us understand that change. Finally, the prediction. The linear regression provides prediction about future values. It is very useful in this regard. This method provides point estimates. This can always prove to be very useful.

Firstly, between two variables the linear approximation is known as simple linear regression. One variable is x , while the other is y . We'll discover the value of y based on x . Moreover, here ' x ' and ' y ' might be anything. For instance, determining the number of fish that survived (y) based on water temperature (x), determining the price of a property (y) based on its location (x), and so on. The next linear equation can be used to express it.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad (4.2)$$

Here, Y_i means Dependent variable, β_0 means intercept, β_1 means slope coefficient and ϵ_i means random error component.

Furthermore, for linearly separable data, linear regression works remarkably well and It is easier to implement, interpret, and train. Furthermore, it's easy to grasp

and models are simple to update with new fresh data.

```
from sklearn.linear_model import LinearRegression
regr = LinearRegression()
regr.fit(X1_train, y1_train)
pred = regr.predict(X1_test)
print(pred)
```

Figure 4.18: Linear Regression.

4.4 Input Data Pre-processing

The data that we are working with has 14 columns with 4080 values. The data is taken from Dhaka US Consulate Air Pollution: Real-time Air Quality Index. However, we only needed the AQI which is mainly the PM2.5 reading, AQI concentrated value, and also AQI concentrated value for 24 hours.

After the download of the dataset, it comes in an excel file. The data will look like Figure 4.17 The attribute for the first 6 columns counting from 0 to 6 is Site, Parameter, Date, Year, Month, Day, and Hour. These columns and attributes are not needed. Therefore, these data will not be detected. Column 7 to 9 is needed which is the actual data and all of the attributes are needed in the proposed model.

Dhaka	PM2.5 - Pr #####	2017	12	1	1	204.2	254	Very Unhe	217	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	2	189.6	240	Very Unhe	175	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	3	177.8	228	Very Unhe	166	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	4	173.9	224	Very Unhe	170	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	5	172.9	223	Very Unhe	169	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	6	160	210	Very Unhe	141	UG/M3	1 Hr	Valid
Dhaka	PM2.5 - Pr #####	2017	12	1	7	160.4	211	Very Unhe	158	UG/M3	1 Hr	Valid

Figure 4.19: A Sample of Raw Data of the collected Dataset.

4.5 Result implementation

After running all of the algorithms we can see that there is an almost accurate result we are getting. In our data set the resulting classifier is stated as “QC Name” which only declares two values that are valid and missing. As air pollution is happening all over the world in every city most of the time the value will be found valid because pollution will be found.

As a result, all of these algorithms show almost accurate values in terms of detecting air pollution. The accuracy graph is shown in 4.20 is generated using google colab and shows accuracy for all the algorithms.

Not to mention, the algorithms used in implementation give us categorical discrete values. But to find our prediction we need continuous values. Therefore, Linear Regression Algorithm has been used.

No	Algorithm	Accuracy
1	Linear Regression	1.0-0.773393773504432
2	Logistic Regression	0.9963235294117647
3	Decision Tree	0.9975490196078431
4	GaussianNB Algorithm	0.9963235294117648
5	Gradient Boosting Algorithm	0.9865196078431373
6	SVC Algorithm	0.9963235294117647
7	Passive Aggressive Classifier	0.9963235294117647
8	SGD Algorithm	0.9963235294117647
9	MLP Classifier Algorithm	0.986516078431373
10	KNN Algorithm	0.9963235294117648
11	Linear SVC	0.9963235294117647
12	Random Forest Algorithm	0.9963235294117647

Table 4.1: Accuracy of Algorithms.

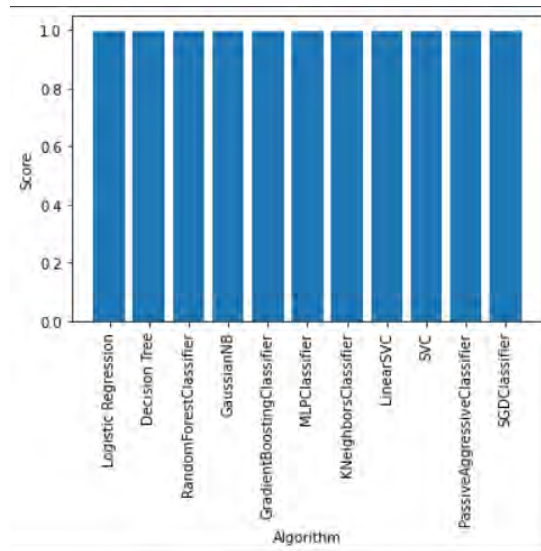


Figure 4.20: Accuracy Graph.

Equally important to talk about the prediction, the Linear Regression algorithm tries to predict the upcoming raw concentration unit of PM2.5 particles per hour and we can see that, it can predict by comparing it with the existing data set. Though its prediction accuracy is high most of the time and sometimes drop a little.

Here in Figure 4.21, we can see the prediction values that we have got by using Linear Regression Algorithm.

Predicting PM2.5 was the primary goal of our work from the beginning but we have also come up with a device that can help us in doing this.

In the figure 4.22 we can see the connection process of the DC motor with the micro-controller.

```
↳ [[ 321.55719595]
    [ 61.30189252]
    [ 185.20717028]
    [ 153.70937687]
    [ 167.09648368]
    [ 124.48288376]
    [ 29.29604874]
    [ 214.65226045]
    [ 237.3956162 ]
    [ 151.65667363]
```

Figure 4.21: Linear Regression Algorithm Prediction Values.

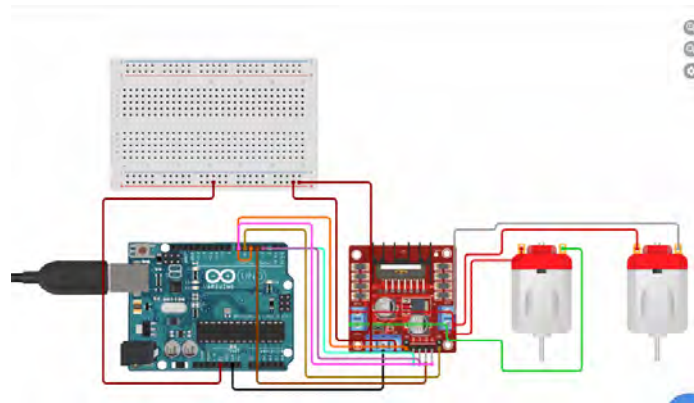


Figure 4.22: Circuit model

We are using a PM2.5 detection sensor which is PMS5003 to get our live PM2.5 value. This live value is collected and also will be collected in future from area where the chances of PM2.5 value is high like industrial zone. The sensor is connected with a Arduino Uno which is already programmed to get data from the sensor. It will give output and the output data will be saved as a csv file. After that the data will be preprocessed for our next use. The data will go through another code for prediction. Data will be trained and tested to get most accurate prediction by using linear regression. The regression will be evaluated by finding MAE, R-Squared and RMSE value. The prediction data will be saved in a datasheet and that saved data will be used by another microcontroller which will control the motor. The work of the motor is to spray water when needed and the microcontroller control the motor according to the predicted value. When prediction gives high value in a specific time period the controller will be set automatically to spray during that specific time. The motor will stop when the value drops to a standard. This prediction-based spraying detects high PM2.5 value before time and can step for the future.

In the figure 4.23 we can see the proposed model of device.

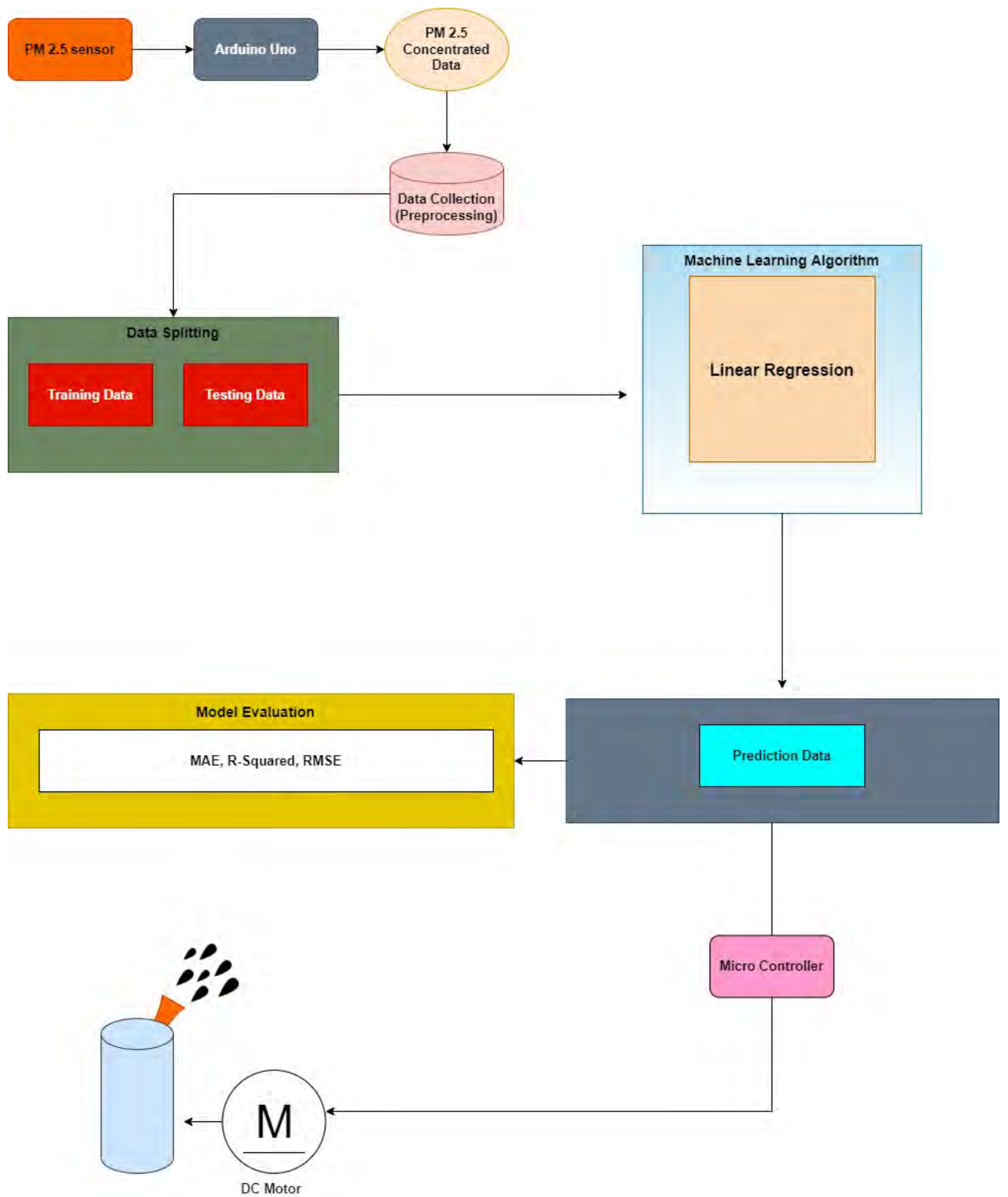


Figure 4.23: Device model

Chapter 5

Limitations and Future work

According to [26], proved that a consequence of the rain and snowy environment increasing source emissions and decreasing atmospheric vertical convection, the concentration of PM_{2.5} is reduced, which improves air quality. The data we have taken from various places like Mohakhali, Bashundhara, and Nabisco shows us that rain indeed decreases the concentration value of PM_{2.5}. In future, we can collect data from many more places including different country to analyze more broadly how the rain or water reduce the PM value. Our device model can be proposed where the main factor will be a water sprayer to reduce the emission of PM_{2.5} particles but our device working method need to updated in future. We have tested twelve algorithm and selected linear regression for prediction. In future, we can test many more algorithm and can find more suitable algorithm for the prediction. As the model is using datasheet from a PC/disk as an input, the device can be as an automated cloud-based system where the device will take data continuously from cloud and calculate and it will also predict. Another future plan that will help people to get serious about the air pollution is the cloud base data can be uploaded to a specific website so that people can be aware of the situation. Lastly, there can be an automated text message or email system for the authority if there is a chance of getting a very high value of PM 2.5. The authority can be any person of that area or any government sector who will use and take the text message or email seriously.

Chapter 6

Conclusion

To conclude, it can be said that, as time goes on air pollution will increase day by day if we don't take steps into our hands. Technology can help us a lot but only to a certain limit. Only we can help ourselves with the help of technology. In a way, as AI can help humans side by side, they are the reflection of human thinking. If we cannot change ourselves then it can be said that AI will bring destruction also. Therefore, by considering the outcome provided by AI we need to take things into our hands to reduce air pollution and that is a hope for a better future.

Bibliography

- [1] B. G. Bennett, J. G. Kretzschmar, G. G. Skland, and H. W. de Koning, “Urban air pollution worldwide,” *Environmental Science & Technology*, vol. 19, no. 4, pp. 298–304, 1985, PMID: 22283338. DOI: 10.1021/es00134a603. eprint: <https://doi.org/10.1021/es00134a603>. [Online]. Available: <https://doi.org/10.1021/es00134a603>.
- [2] G. W. Taylor and M. P. Becker, “Increased efficiency of analyses: cumulative logistic regression vs ordinary logistic regression,” *Community Dentistry and Oral Epidemiology*, vol. 26, no. 1, pp. 1–6, 1998. DOI: 10.1111/j.1600-0528.1998.tb02075.x.
- [3] A. Prochazka, M. Kolinova, J. Fiala, P. Hampl, and K. Hlavaty, “Satellite image processing and air pollution detection,” in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 4, 2000, 2282–2285 vol.4. DOI: 10.1109/ICASSP.2000.859295.
- [4] H. Akimoto, “Global air quality and pollution,” *Science*, vol. 302, no. 5651, pp. 1716–1719, 2003. DOI: 10.1126/science.1092666. eprint: <https://www.science.org/doi/pdf/10.1126/science.1092666>. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.1092666>.
- [5] A. Ben-Hur, *Support vector clustering - Scholarpedia*, Jun. 2008. [Online]. Available: http://www.scholarpedia.org/article/Support_vector_clustering.
- [6] J. W. Park, C. H. Yun, H. S. Jung, and Y. W. Lee, “Visualization of urban air pollution with cloud computing,” in *2011 IEEE World Congress on Services*, 2011, pp. 578–583. DOI: 10.1109/SERVICES.2011.111.
- [7] B. de Ville, “Decision trees,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 6, pp. 448–455, 2013. DOI: 10.1002/wics.1278.
- [8] X. Xi, Z. Wei, R. Xiaoguang, *et al.*, “A comprehensive evaluation of air pollution prediction improvement by a machine learning method,” in *2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*, 2015, pp. 176–181. DOI: 10.1109/SOLI.2015.7367615.
- [9] *AQI Calculations Overview- Ozone, PM2.5 and PM10*, Jul. 2016. [Online]. Available: <https://forum.airnowtech.org/t/aqi-calculations-overview-ozone-pm2-5-and-pm10/168>.
- [10] P. D. Kaur and P. Singh, “Optimization of cloud resources for air pollution monitoring devices,” in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, 2016, pp. 1–6.

- [11] *Air pollution*, Aug. 2018. [Online]. Available: <https://www.who.int/westernpacific/health-topics/air-pollution>.
- [12] R. O. Sinnott and Z. Guan, “Prediction of air pollution through machine learning approaches on the cloud,” in *2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)*, 2018, pp. 51–60. DOI: 10.1109/BDCAT.2018.00015.
- [13] B. Guanochanga, R. Cachipuendo, W. Fuertes, *et al.*, “Real-time air pollution monitoring systems using wireless sensor networks connected in a cloud-computing, wrapped up web services,” in *Proceedings of the Future Technologies Conference (FTC) 2018*, K. Arai, R. Bhatia, and S. Kapoor, Eds., Cham: Springer International Publishing, 2019, pp. 171–184, ISBN: 978-3-030-02686-8.
- [14] T. Sayahi, A. Butterfield, and K. Kelly, “Long-term field evaluation of the plantower pms low-cost particulate matter sensors,” *Environmental Pollution*, vol. 245, pp. 932–940, 2019, ISSN: 0269-7491. DOI: <https://doi.org/10.1016/j.envpol.2018.11.065>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0269749118316129>.
- [15] DataTechNotes, *Classification Example with Linear SVC in Python*, Jul. 2020. [Online]. Available: <https://www.datatechnotes.com/2020/07/classification-example-with-linearsvm-in-python.html>.
- [16] GeeksforGeeks, *Passive Aggressive Classifiers*, Jul. 2020. [Online]. Available: <https://www.geeksforgeeks.org/passive-aggressive-classifiers/>.
- [17] A. Joshi, *Do you know plants that give oxygen 24 hours ?* Mar. 2020. [Online]. Available: <https://nurserylive.com/blogs/sustainable-living/do-you-know-plants-that-give-oxygen-24-hours>.
- [18] P. Majumder, *Gaussian Naive Bayes*, Feb. 2020. [Online]. Available: <https://iq.opengenus.org/gaussian-naive-bayes/>.
- [19] Corporate Finance Institute, *Random Forest*, Sep. 2021. [Online]. Available: <https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/>.
- [20] GeeksforGeeks, *ML — Stochastic Gradient Descent (SGD)*, Sep. 2021. [Online]. Available: <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>.
- [21] V. Kurama, *Gradient Boosting for Classification*, Apr. 2021. [Online]. Available: <https://blog.paperspace.com/gradient-boosting-for-classification/>.
- [22] *Machine Learning Random Forest Algorithm - Javatpoint*, 2021. [Online]. Available: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>.
- [23] A. Masood and K. Ahmad, “A review on emerging artificial intelligence (ai) techniques for air pollution forecasting: Fundamentals, application and performance,” *Journal of Cleaner Production*, vol. 322, p. 129 072, 2021, ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2021.129072>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959652621032613>.

- [24] E. Nizeyimana, D. Hanyurwimfura, R. Shibasaki, and J. Nsenga, “Design of a decentralized and predictive real-time framework for air pollution spikes monitoring,” in *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 2021, pp. 501–504. DOI: 10.1109/ICCCBDA51879.2021.9442611.
- [25] S. Sharma, *KNN - The Distance Based Machine Learning Algorithm*, May 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/05/knn-the-distance-based-machine-learning-algorithm/>.
- [26] X. Tian, K. Cui, H.-L. Sheu, Y.-K. Hsieh, and F. Yu, “Effects of rain and snow on the air quality index, PM2.5 levels, and dry deposition flux of PCDD/fs,” *Aerosol and Air Quality Research*, vol. 21, no. 8, p. 210 158, 2021. DOI: 10.4209/aaqr.210158. [Online]. Available: <https://doi.org/10.4209%2Faaqr.210158>.
- [27] Editorial Team, *The Benefits of AI-Assisted Software Development*, Feb. 2022. [Online]. Available: <https://sparkequation.com/2020/12/10/ai-assisted-development/>.
- [28] The World Air Quality Index project, *World’s Air Pollution: Real-time Air Quality Index*, May 2022. [Online]. Available: <https://waqi.info/#/c/7.984/8.91/2.3z>.
- [29] *Types of air pollution - British Lung Foundation*, Feb. 2022. [Online]. Available: <https://www.blf.org.uk/support-for-you/air-pollution/types>.
- [30] *Scikit Learn - Stochastic Gradient Descent*. [Online]. Available: [https://www.tutorialspoint.com/scikit_learn/scikit_learn_stochastic_gradient_descent.htm#:~:text=Stochastic%20Gradient%20Descent%20\(SGD\)%20is,as%20SVM%20and%20Logistic%20regression.](https://www.tutorialspoint.com/scikit_learn/scikit_learn_stochastic_gradient_descent.htm#:~:text=Stochastic%20Gradient%20Descent%20(SGD)%20is,as%20SVM%20and%20Logistic%20regression.)