

AIR POLLUTION CONTROLLER SYSTEM USING MACHINE LEARNING AND IOT

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

PEDDIREDDY ANILKUMAR (Reg.No:37110558)

PACHIGOLLA SURENDRA (Reg.No:37110526)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119**

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **PEDDI REDDY ANILKUMAR (Reg. No. 37110558)** and **PACHIGOLLA SURENDRA (Reg. No. 37110526)** who carried out the project entitled “**AIR POLLUTION CONTROLLER SYSTEM USING MACHINE LEARNING AND IOT**” under my supervision from **August 2020** to **March 2021**.

Internal Guide Name

Dr.L.SUJI HELEN., M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **PEDDIREDDY ANILKUMAR(Reg.no.37110558)** hereby declare that the Professional training Report entitled **“AIR POLLUTION CONTROLLER SYSTEM USING MACHINE LEARNING AND IOT”** done by me under the guidance of **Dr.L.SUJIHELEN.,M.E.,Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA., M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI M.E., Ph.D., and Dr. L. LAKSHMANAN M.E., Ph.D.**, Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L.SUJIHELEN.,M.E.,Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The effect of hurtful toxins in the air on human well being is an immense territory of exploration, forestalling or Controlling, and furthermore checking the poison is the gigantic obligation of any administering body. The meteorological and traffic factors, consuming of petroleum derivatives, mechanical boundaries, for example, power plant discharges assume huge parts in air pollution. Among all the particulate matter that decide the nature of the air PM 2.5 needs more consideration. When its level is high noticeable all around, it causes difficult issues on individuals' well being. Thus, controlling it by continually keeping a beware of its level noticeable all around is significant. This paper deals with measuring the Air Quality using MQ135 sensor along with Carbon Monoxide CO using MQ7sensor. Measuring Air Quality is an important element for bringing awareness to take care of the future generations and for a healthier life. Machine Learning analysis brings us a lot of depth in understanding the information that we obtained from the data. Moreover, we are providing a reduction of the cost of components versus the state of the art.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGUERS	iv
	LIST OF ABBREVATIONS	v
1	INTRODUCTION	1
	1.1 INTRODUCTION OF PROJECT	1
	1.2 PURPOSE	2
	1.3 MOTIVATION	3
2	LITERATURE SURVEY	4
	2.1 ANALYSIS OF THE LITERATURE	4
	2.2 LITERARY REVIEWS	5
3	AIM & SCOPE OF THE PRESENTINVESTIGATION	8
	3.1 IOT DEVICE PROTOTYPE	8
	3.2 PREDICTIVE MODELING	12
	3.2.2 ADVANTAGES	8
4	EXPERIMENTAL OR MATERIALS AND METHODS	17
	ALGORITHMS USED	
	4.1 MACHINE LEARNING ALGORITHM	17
	4.2 IOT ALGORITHMS USED	18
	4.3 REQUIREMENTS ANALYSIS	19
	4.3.1 SOFTWARE REQUIREMENTS	

	4.3.2 HARDWARE REQUIREMENTS	20
	4.4 SYSTEM FUNCTIONALITIES	20
	4.4.1 FUNCTIONAL REQUIREMENT	21
	4.4.2NON FUNCTIONAL REQUIREMENT	22
	4.5 TECHNOLOGY USED	22
	4.5.1 AURDINO	25
	4.5.2 PYTHON	26
	4.6 MODULES	27
	4.6.1 NUMPY	29
	4.6.2 PANDAS	30
	4.6.3 MATPLOTLIB	31
	4.6.4 DATETIME	32
	4.6.5 SEABORN	
5	RESULTS AND DISCUSSION	33
	5.1 RESULT ANALYSIS	33
6	SUMMARY AND CONCLUSION	36
	REFERENCES	37

APPENDIX	39
A. PLAGIARISM REPORT	39
B. JOURNAL PAPER	40
C. SCREEN SHOTS	50
D. SOURCE CODE	56

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1.1	AIR POLLUTION MONITORING SYSTEM	8
3.1.2	NODE MCU BOARD WITH PIN	10
3.1.3	MQ-135 GAS SENSOR	10
3.1.4	FLOW CHART DIAGARM	10
3.1.5	IMPLMENTING PROTOTYPE	11
3.1.6	THINK SPEAK CLOUD DATA COLLECTION	11
3.2	SOLUTION ARCHITECTURE IN PREDICTIVE	16
4.1	DATA FLOW	18
4.5.1	MQ-135 GAS SENSOR	24
4.5.2	NODE MCU ESP8266	25
5.1	AIR POLLUTION COLLECTION DATA	33
5.2	HORLY POLLUTION LEVEL AT DRAKSHARAM	34
5.3	HOURLY POLLUTION LEVEL AT RCPM	34
5.4	HOURLY POLLUTION LEVEL AT PASALAPUDI	35

CHAPTER 1

INTRODUCTION

1.1 Introduction

It has been observed that in industrially developing or developed countries human health get majorly affected due to Air Pollution, where there is no infrastructure to monitor it or keep it under control .

In recent researches it has been proven that there is a high correlation between atmospheric pollutants and diseases like asthma and lung related diseases. Air pollution is now a major concern across the globe and World Health Organization (WHO) has developed certain guidelines to restrict the limits of certain gases like Ozone (O₃), Nitrogen Dioxide (NO₂), Sulfur Dioxide (SO₂). The Air Quality Index calculation and pollution monitoring are mainly done Air Quality

monitoring (AQM) stations that are pretty much precise and accurate. They exhibit optimum reliability and effective in measuring a wide range of air pollutants. But even after all these stations lag mainly in three areas: Infrastructure, necessary for installation because of the huge size, Operational requirements are pretty much complex, The recurring expenses of setting up, daily maintenance and calibration. As a result of these flaws, the numbers of installations are less and eventually lead to discrete AQM Networks with limited air pollution data to collect .

The recent advancements in embedded electronics have enabled the usage of wireless network technologies in monitoring different real-time sensor data, like temperature, pressure and air pollution. The data retrieved by the devices are sent wirelessly to a database on a remote server for future analysis and processing. The concept of usage of small size, and cheap AQM motes have certainly increased mainly because of the wireless transferring of data .

The aim in this paper is to find out the best prediction and forecasting model for rise or fall of the certain air pollutants like O₃, NO₂, SO₂, and CO which are all considered harmful according to WHO guidelines. Three Machine Learning algorithms have been

implemented which include, Linear Regression, Random Forest, XG Boost and ARIMA for time-series forecasting.

All of these models have good predictive capacity along with generalization power and have a wide range of applications. This paper has been organized as follows: Section I gives the introduction and related machine learning models.

Section II deals with the proposed IoT based Air pollution monitoring system and proposed predictive and forecasting modeling on an open-source dataset.

Section III deals with the result analysis of the pollution data in the context of predictive ML model in the context of various forecasting models.

Section IV concludes the research

1.2 PURPOSE

The purpose of this proposed system i.e. Air pollution controller system using IoT and machine learning Air quality is a critical issue that straightforwardly influences human well being. Air quality information are gathered remotely from checking bits that are outfitted with a variety of vaporous also, meteorological sensors.

This information are investigated and utilized as a part of anticipating fixation estimations of contamination utilizing savvy machine to machine stage. The stage comprises of a ML-based calculations to construct the estimating models by training from the gathered information.

However we can conclude that we can use gradient boosting method for prediction, preferably XG Boost because of its level-wise approach and helps in building a model which has low bias and low variance. ARIMA performs pretty well as a forecasting model, what can be used as to make a daily forecast just like regular weather forecasting.

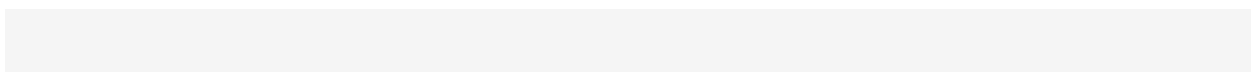
monitoring (AQM) stations that are pretty much precise and accurate. They exhibit optimum reliability and effective in measuring a wide range of air pollutants. But even after all these stations lag mainly in three areas:

Infrastructure, necessary for installation because of the huge size, Operational requirements are pretty much complex, The recurring expenses of setting up, daily

maintenance and calibration. As a result of these flaws, the numbers of installations are less and eventually lead to discrete AQM Networks with limited air pollution data.

1.3 MOTIVATION

The purpose of this proposed system i.e. Air pollution controller system using IoT and machine learning is working towards The rapid growth of industry and transport within this contemporary progress, there was sufficient consideration given to air quality monitoring; but conventional air quality monitoring methods are inefficient to produce adequate spatial and temporal resolutions of the air quality information by cost-effective also the period time clarifications. During the paper, we propose a distinct methodology to achieve the air quality monitoring system, using this fog computing-based Internet of Things (IoT). In this paper proposed an embedded system, where sensors collect the air quality information within period time and send it over the fog nodes. Every fog node may be an extraordinarily visualized program hosted at a committed computing node implemented with a connection interface. Data gathered by Microprocessor based IoT sensing things do not seem to be causing on into the cloud server to the process. Preferably, they do send through the adjacent fog node to get quick, including high-rise rate service. Though, fog node will refine non-actionable data (e.g., regular device measurement) also forward them to the Cloud for lengthy run storage and batch analysis. The Cloud may be a convenient location to run world analysis at information gathered from commonly shared devices over sustained periods (months, years). General purpose processor (microprocessor) and IoT cloud platforms were involved in developing this whole infrastructure and model for analysis. Empirical outcomes reveal that this advanced method is responsible for sensing air quality, which serves to expose the modification patterns regarding air quality through a certain level.



CHAPTER 2

LITERATURE SURVEY

2.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide the time factor, economy and friends strength. When these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Prior to building the framework the above thought are considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friends strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in get assets from different areas and all the connected points that are exceptionally valuable under this segment. The most awesome aspect if this is the manner in which things get all together and encourages us to suite our work according to the current information.

2.2 LITERARY REVIEWS

“Machine Learning Approach for Air Quality Prediction” Dixian Zhu(2018)

Utilized the Bias organization to discover the air quality and framed DAG from the informational collection of the town called as shanghai. The data set is plunged for the preparation and testing model. The inconvenience of this methodology is they have not thought about topographical and social climate attributes, so the outcomes may change dependent on these variables.

“Air pollution characteristics and their relation” J. He, S. Gong(2016)

Introduced the most awesome aspect our insight, the nearest to our work is multi scale PM2.5 expectation utilizing long momentary memory neural organization expanded model. His contend that current measurable and AI models for foreseeing air poison fixation neglect to successfully display long haul conditions. Furthermore, it was additionally indicated that the utilization of assistant information improved model execution.

“Empirical Study of PM2.5 Forecasting Using neural network”

Sachet Mahajan(2017)

Proposed fake based Regressive model which is nonlinear to foresee 2 significant air toxins 2, for example, carbon monoxide and nitrogen oxides. They have considered the factors, for example, the speed of the air, air bearing, temperature, and dampness and the harmful components from the modern site, for example, Skikda. They have utilized RMSE and MAE to decide the exhibition, yet this strategy considered just 2 toxins, for example, NO and CO the other significant pollutions, for example, sulfur dioxide, PM2.5, PM10 are not thought of.

“Machine learning techniques in air quality prediction” E. Kalapanidas(2017)

Actualized the possibility of with less expense i.e., while pushing the information to the Cloud, there is no compelling reason to see the yield on LCD which adds more expense to the. Focusing on IoT as a stage, our in tension ought to be to introduce the information on Internet utilizing the stages like thinker.io or thing speak or Cayenne site which are delightfully intended to introduce the yield and even ready to download the data set. While doing an analysis for air quality observing, there is no compelling reason to utilize LPG (Liquified Petroleum Gas) or methane identifying sensors as it is utilized for home/office well being. This paper utilized WiFi to push the information onto the

Cloud rather utilizing GSM or GPRS module as in.

“Machine learning techniques on air quality data for realtime” Athanasiadis(2015)

As this paper utilizes two sensors, the two of them have interior warmer component, it draws more force ($P=V*I$), so however the two sensors are turned ON, its yield voltage levels change and shows erratic qualities because of deficient force drive. So, we utilized a 9 Volts battery and a 7805 family LM7805 Regulator for the CO sensor MQ7 as the force from Arduino alone isn't adequate to drive two sensors.

“A Simple Feedforward Neural Network for the PM10 Forecasting” Caselli(2016)

Isn't clear with the parts utilized and the cloud utilized. This paper additionally plans to actualize the AI on the continuous data set gathered from the Thing speak site and attempt to pass on the data about the unfavorable impacts on one's well being to individuals and government if a similar contamination proceeds. This paper additionally plans to reach out by adding three additional sensors identified with air quality, i.e, ozone sensor, PM10 laser dust sensor, MG811 (CO₂) sensor that goes about as an across-the-board arrangement giving inside and out observing of the air quality.

“The lot architecture, possible applications and key challenges” Khan (2014)

Had totally taken wrong supposition where they have showed the yield 997 PPM as the natural air, while Delhi which is the most dirtied city recording 250 PPM. Its unmistakable arrangement that they have not adjusted the sensor and didn't change over the crude sensor information into PPM utilizing determinations we did. They have utilized a Local Host which is restricted where they can see the yield just on the PC inside the territory where trial arrangement is associated. However, this paper targets utilizing standard IoT stage which is exceptionally gotten and open source.

“ThingSpeak based sensing and monitoring system” Pasha(2016)

Proposed an IoT based method to acquire air quality informational collection. They have utilized Long Short-term Memory [LSTM] procedure to foresee the air quality the proposed method accomplished better precision by decreasing the time taken to prepare the model. Yet at the same time, the exactness can be improved by looked at different strategies.

“IoT enabled environmental monitoring system for smart cities” Shah(2016)

described the benefits of the Bidirectional Long - Short `Memory [BiLSTM] method to forecast the severity of air pollution. The proposed technique achieved better prediction

which models the long term, short term, and critical consequence of PM_{2.5} severity levels. In the proposed method prediction is made at 6h, 12h, 24h. The results obtained

for 12h is consistent, but the result obtained for 6h, and 24h are not consistent. Actualized the possibility of with less expense i.e., while pushing the information to the Cloud, there is no compelling reason to see the yield on LCD which adds more expense to the. Focusing on IoT as a stage, our intension ought to be to introduce the information on Internet utilizing the stages like thinger.io or thingspeak or Cayenne site which are delightfully intended to introduce the yield and even ready to download the dataset. While doing an analysis for air quality observing, there is no compelling reason to utilize LPG (Liquified Petroleum Gas) or methane identifying sensors as it is utilized for home/office wellbeing.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 IOT DEVICE PROTOTYPE

IoT mainly deals with connecting smart devices (embedded electronics devices) to internet by harnessing the advantage of OSI layered Architecture. In the context of this work we propose a cluster of Air Quality Monitoring Sensor motes, which are used to measure the concentration of Air pollutants in the air.

All the Air Sensors are interfaces with a tiny embedded platform equipped with network connectivity and are interconnected to internet making it a global network of connected things . We have mainly used the NodeMCU which is an open source development boards with ESP8266-12E chips

MQ-135 Gas Sensor is used to collect gas concentration measurements. This sensor data would be captured and sent to the Thing Speak cloud for IoT based data acquisition. Fig.3.1.1 shows the proposed system architecture.

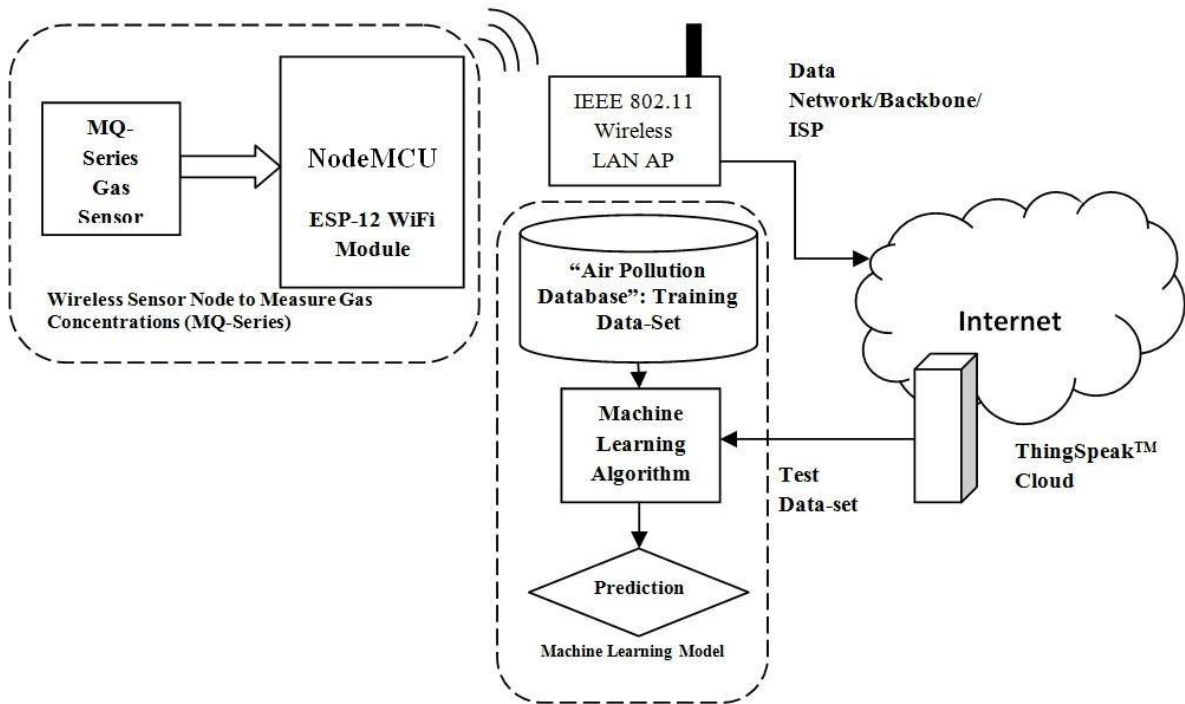


Fig. 3.1.1: Block diagram of proposed IoT based Air Pollution Monitoring System

Fig.3.1.2 shows the pin configuration of NodeMCU Board which is used to interface the MQ2 gas sensor, ESP8266 WLAN adaptor is embedded inside the device. Fig.3.1.3 is the MQ-135 Gas Sensor, which is used in this work to get the different concentration of gas.

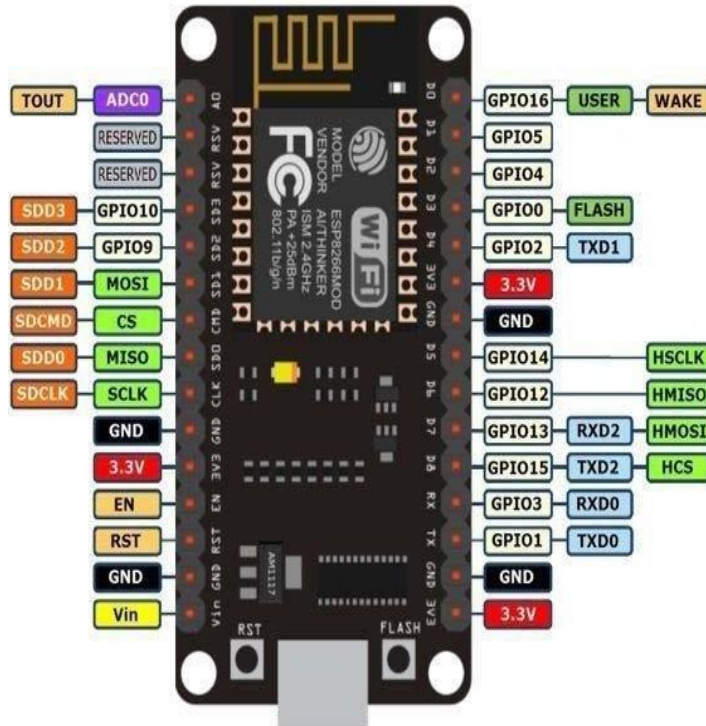


Fig. 3.1.2 Node MCU Board with PIN configuration



Fig. 3.1.3 MQ-135 Gas Sensor

Fig.3.1.4 depicts the flowchart for IoT based Air pollution monitoring and forecasting using Predictive Machine Learning model

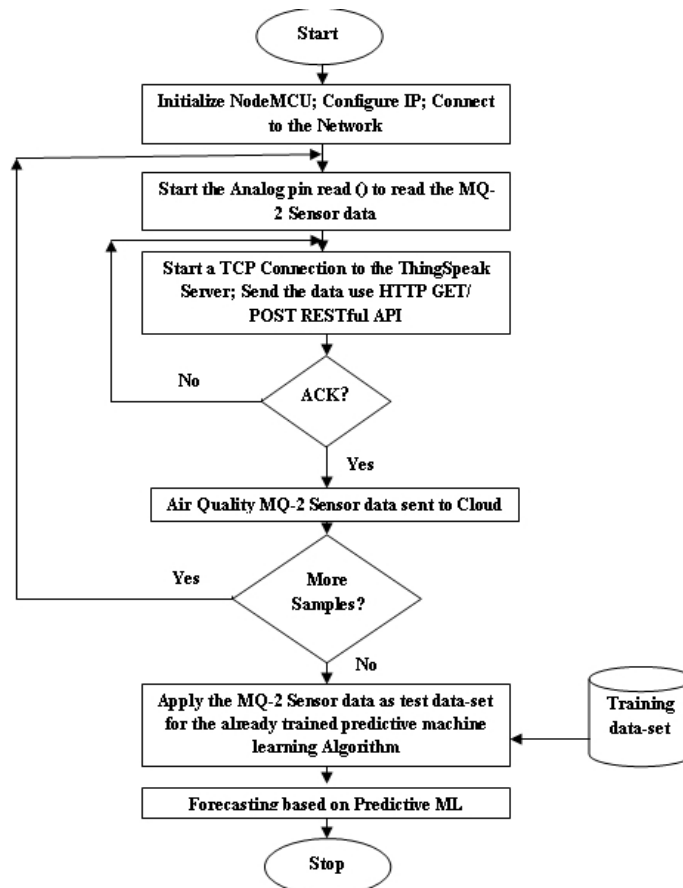


Fig. 3.1.4 Flowchart for the IoT based Air Pollution Monitoring System

Further data processing and use of machine learning model can be done on the cloud but not on the device to reduce the computational overheads of tiny embedded devices as they are constrained low power devices and mostly powered by battery.

Reducing computational complexity of these sensor motes would drastically increase the operational time by optimizing the battery life and in addition to that cheaper deployments as simpler SoC and TinyOS firm factor devices can be deployed. Fig.3.1.5 shows the implementation of the proposed device to monitor air quality. The figure shows the prototype with connected NodeMCU with the MQ-2 gas sensor interfaced on the breadboard.



Fig.3.1.5: Implementing Prototype

In this work the ThingSpeak has been used as the cloud and visualization, which is an open source cloud platform to collect our data. Fig.3.1.6 shows the Thing Speak output screenshots of the data visualization.

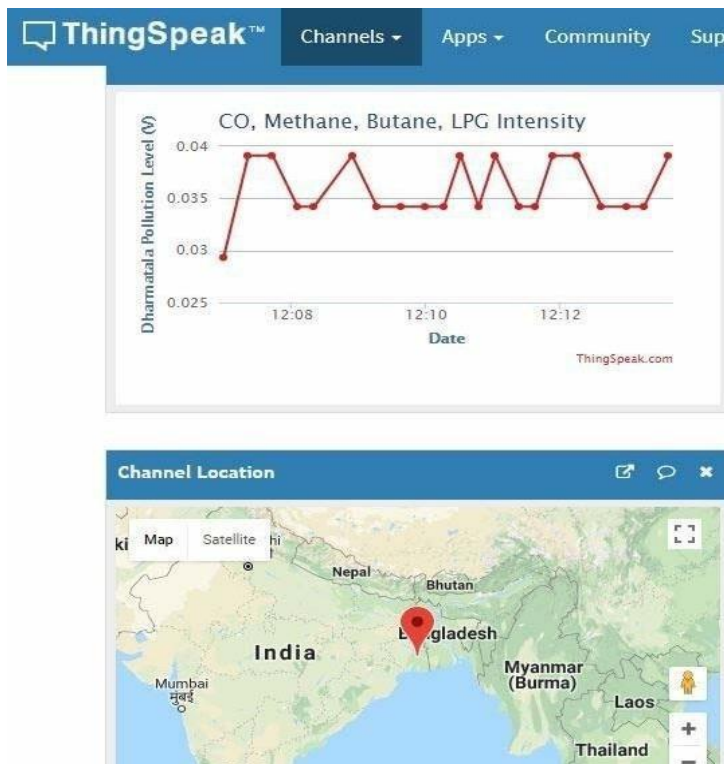


Fig. 3.1.6: ThingSpeak Cloud Data Collection

3.2 PREDICTIVE MODELING

ML mainly deals with computational methods that enhance the execution of automating the securing of learning from encounter [16]. The process of learning by a machine from complex set of data and solving critical problems, being more intelligent is what machine learning is all about.

Just like, there's a regular weather forecasting done for the next day, in the same way the pollution forecasting model can be used so that people can take precautionary measures.

We aim to accurately predict concentrations of O₃, SO₂, NO₂ and CO. For testing our model we have used an authorized pollution dataset provided by US EPA, which consists of pollution data for 46 states of the USA over the last 15 years. The process of building the prediction models mainly deal with 3 steps:

- 1) Data Pre-processing: The first step of building a prediction model is data pre-processing where data is cleaned, missing values are filled, outliers are removed and also data is arranged in a way to fit for the Machine Learning algorithms.
- 2) Feature Engineering: Features are one of the major factors which increase the prediction accuracy such, day, month, time of the day, etc.
- 3) Building Forecasting Model: Model is built to predict the future, i.e. on the unseen data based on the historical data In training data know target variables are stored, features are selected and these features are used in all the algorithms.

For the testing part the developed models are cross-validated and evaluated. The model was evaluated using cross-validation techniques based on Root mean square error (RMSE) and the mean absolute error (MAE) .

The models are implemented and their individual performance is evaluated in this work, which gives a qualitative measure of model's performance. Mean Absolute Error (MAE) is the measurement of the average magnitude of the errors in a set of predictions is calculated without considering their direction.

MAE is the average of absolute difference of the entire test sample between the actual observation and the prediction data set, here all individual difference have equal

weight as given in equation .

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (1)$$

Root Mean Square Error (RMSE) is quantitative measurements of difference between the values those are actually observed and the values (sample and population values) predicted by a model or an estimator. RMSE is given in equation (2) as follows

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}} \quad (2)$$

For implementing the model in real-life the best features and prediction model will be used for the unseen data. In the volatile environmental changes the tautology of training and testing and deploying will be done periodically.

This is an iterative process which should be done to improve the model performance. There are numerable algorithms that can be implemented, but out of which Random Forest and XGBoost proved to be the most successful models.

Linear Regression is used in machine learning for a kind of perceptive analysis. Two major factors are to be taken into account while applying Linear Regression in ML.

whether an arrangement of indicator factors is good enough in a distinct visibility with reference to foreseeing a result (subordinate) variable?

Which factors typically are dominant indicators of the variable obtained through data acquisition, and how do they— established by the greatness and indication of the beta estimates— affect the variable? However the regression measure shows the connection between one ward variable and at least one of the free factors. The easiest type of the relapse condition with one needy and one free factor is characterized by the equation $y = c + bx$, where y = assessed subordinate variable score, c = consistent, b = regression coefficient, and x = score on the independent variable.

Random Forest is an adaptable, simple to utilize machine learning calculation that produces, even without hyper parameter tuning, an awesome outcome more often than not. It is additionally a standout among the most utilized calculations, since its effortlessness and the

way that it can In the IA technique, multiple transmitters jointly encode their signals to intended receivers such that interfering signals are separated and eliminated.

Simulation results demonstrate that compared to TDMA algorithms, which increase the performance of the network delay and throughput.

An experimental wireless system that extends beyond the hot spots capabilities to provide wireless connectivity at distant areas and at a low cost. The system combines the paradigm of Wireless Mesh Networks with the Captive Portal technology to supply a large varies of Internet-based communication services and applications.

Finally, some experiments in terms of context services and traffic modeling, and demonstrate that the developed system can be easily deployed in terms of coverage, management, and offered services. Explores 5 key challenges during this field and supply a response through a detector information platform which might take detector information from assortment to final product via a data repository and workflow system.

This will activate the rapid development of applications built on sensor data using data fusion and the integration. Gradient boosting is a machine learning (ML) technique typically used for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, based on a decision trees.

A stage-wise fashion model is built by GBM very similar to other boosting methods. An arbitrary differentiable loss function is used to further generalize them by allowing optimization. The industrial pollution is controlled using the monitored data. But there is a problem in storing big data. There are several methods are used for real- time data storage. The Clean Wi-Fi network constantly monitors the air for pollutant gases, uses that information to feed a Big Sensor Data system, and uses the same data for the automatic configuration of the public Wi-Fi service.

be utilized for both grouping and relapse errands. In this context, it can be inferred the appropriateness and usability of random forest calculation and a few other vital things about it.

XGBoost is another way to say "Extreme Gradient Boosting". The expression "Inclination Boosting" is proposed as a Greedy Function Approximation: A Gradient Boosting Machine, by Friedman. Gradient boosting is currently one of the most popular techniques for efficient modeling of tabular datasets of all sizes. XGBoost is a very fast, scalable implementation of gradient boosting.

XG Boost depends on this unique model. XG Boost is an implementation of gradient boosted decision trees designed for speed and performance. The Gradient Boosting Model (GBM) based on decision tree is a popular machine learning technique.

Gradient boosting is a machine learning (ML) technique typically used for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, based on a decision trees.

A stage-wise fashion model is built by GBM very similar to other boosting methods. An arbitrary differentiable loss function is used to further generalize them by allowing optimization.

Auto regressive integrated moving normal (ARIMA) is one of the well known linear models in time arrangement determining amid the previous three decades. Late research exercises in anticipating with artificial neural systems (ANNs) recommend that ANNs can be a promising contrasting option to the customary linear methods.

ARIMA models and ANNs are regularly contrasted and blended conclusions regarding the prevalence in forecasting models. Fig.7 shows the proposed system architecture, which interprets the training and testing data set. The proposed architecture includes the pre processing and feature extraction with cross validation techniques.

XG Boost depends on this unique model. XG Boost is an implementation of gradient boosted decision trees designed for speed and performance. The Gradient Boosting Model (GBM) based on decision tree is a popular machine learning technique.

This is an iterative process which should be done to improve the model performance. There are numerable algorithms that can be implemented, but out of which Random Forest and XGBoost proved to be the most successful models.

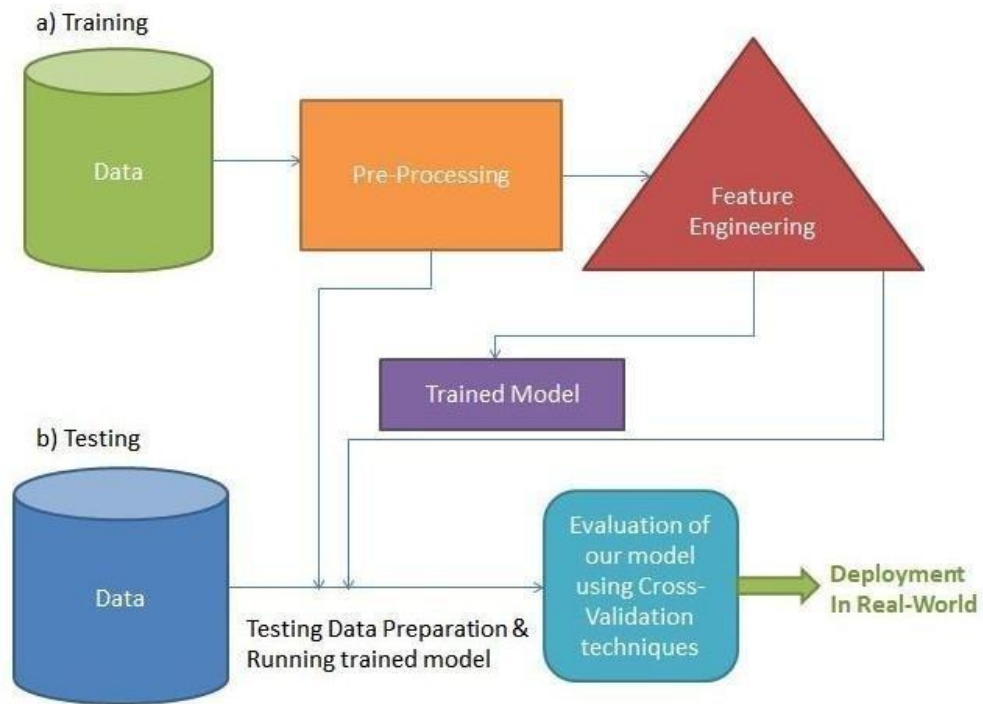


Fig. 3.2 : Solution Architecture in predictive model

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS

ALGORITHMS USED

4.1 Machine learning algorithms used

Modern studies in the field of environment science and engineering show that deterministic models struggle to capture the relationship between the concentration of atmospheric pollutants and their emission sources. The recent advances in statistical modeling based on machine learning approaches have emerged as solution to tackle these issues.

It is a fact that, input variable type largely affect the performance of an algorithm, however, it is yet to be known why an algorithm is preferred over the other for a certain task.

The work aims at highlighting the underlying principles of machine learning techniques and about their role in enhancing the prediction performance. The study adopts, 38 most relevant studies in the field of environmental science and engineering which have applied machine learning techniques during last 6 years.

The review conducted explores several aspects of the studies such as:

- the role of input predictors to improve the prediction accuracy;
- geographically where these studies were conducted;
- the major techniques applied for pollutant concentration estimation or forecasting;
- whether these techniques were based on Linear Regression, Neural Network, Support Vector Machine or Ensemble learning algorithms.

The results obtained suggest that, machine learning techniques are mainly conducted in continent Europe and America.

Furthermore a factorial analysis named multi-component analysis performed show that pollution estimation is generally performed by using ensemble learning and linear regression based approaches, whereas, forecasting tasks tend to implement neural networks and support vector machines based algorithms.

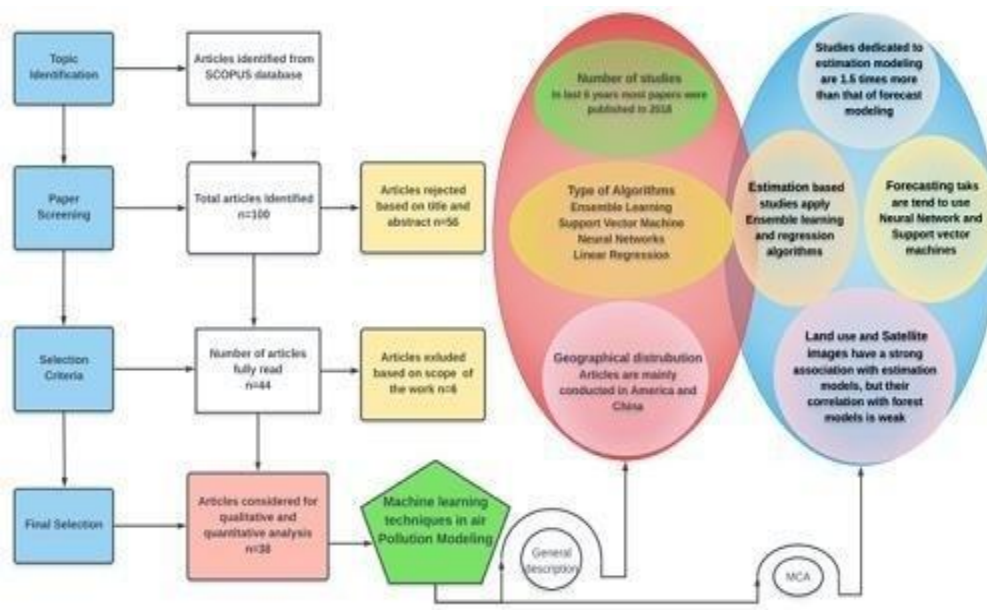


Fig 4.1: Data flow

4.2 lot algorithms used

The industrial pollution is controlled using the monitored data. But there is a problem in storing big data. There are several methods are used for real- time data storage. The Clean Wi-Fi network constantly monitors the air for pollutant gases, uses that information to feed a Big Sensor Data system, and uses the same data for the automatic configuration of the public Wi-Fi service, displaying information about the quality of the air to the user, and rewarding less Polluted areas with a better service.

That way it raises public awareness about the state of air pollution and how important it is to reduce it, promotes the use of renewable energies and brings Wi-Fi connectivity to the people.

But using more electronic components may consume more electricity. Propose a new method that uses an interference alignment technique to mitigate interference effects in Wireless Sensor Networks. It overcomes the

challenge arises in forming a data gathering network to maximize the network capacity.

In the IA technique, multiple transmitters jointly encode their signals to intended receivers such that interfering signals are separated and eliminated. Simulation results demonstrate that compared to TDMA algorithms, which increase the performance of the network delay and throughput.

An experimental wireless system that extends beyond the hotspots capabilities to provide wireless connectivity at distant areas and at a low cost. The system combines the paradigm of Wireless Mesh Networks with the Captive Portal technology to supply a large varies of Internet-based communication services and applications.

Finally, some experiments in terms of context services and traffic modeling, and demonstrate that the developed system can be easily deployed in terms of coverage, management, and offered services. Explores 5 key challenges during this field and supply a response through a detector information platform Concinnity which might take detector information from assortment to final product via a data repository and workflow system.

This will activate the rapid development of applications built on sensor data using data fusion and the integration.

4.3 REQUIREMENT ANALYSIS

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer"s machines already. So the main need is to introduce appropriate equipment for the product. The requirements are split into two categories

4.3.1 SOFTWARE REQUIREMENT

The basic software requirements to run the program are:

1. Operating System: Microsoft Windows XP, Winows 7, Windows 9, Windows 10
2. Software: Python 3.4 & above 4.4.2

4.3.2 HARDWARE REQUIREMENT

1. Hard Disk:500GB & above
2. RAM:4GB & above
3. Processor:i3 & above13

4.4 SYSTEM FUNCTIONALITIES

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.4.1 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

i. Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

ii. Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

iii. Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

iv. Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and even capable of working around any failure

V. Compatibility

It is supported by version above all web browsers. Using any web servers like local host makes the system

Vi. Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users. vii. Safety Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

i.Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

ii. Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

iii. Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective. iv. Maintainability Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.5 TECHNOLOGY USED

4.5.1 ARDUINO

Arduino was a project started at Interaction Design Institute Ivrea (IDII) in Ivrea, Italy, with its primary goal being creating affordable and straightforward tools for non-engineers to use and create digital projects. During its infancy, the project consisted of just three members- Hernando Barragán, Massimo Banzi, and Casey Reas. Hernando Barragán worked under the guidance of Massimo Banzi and Casey Reas and created a development platform called Wiring as his masters' thesis project at IDII. The development platform consisted of the ATmega168 microcontroller as its brains and used an IDE based on Processing, which was co-created by Casey Reas. Later, Massimo Banzi, along with two other students from IDII, namely- David Mellis and David Cuartielles, added support for the cheaper ATmega8 microcontroller. The three, instead of working on developing and improving Wiring, they forked it and renamed the project to Arduino. The initial core Arduino team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, but Barragán was not included.

Now that you know the origin of Arduino, it is essential to get yourself acquainted with the hardware that Arduino as a company offers. One of the main reasons for Arduino being so accessible and affordable across the globe is because all of the Arduino hardware is open-source. Being open-source has a plethora of advantages- anyone can access the design and build of the device and make improvements; anyone can use the same hardware design to create their product lineup. Since Arduino is open- source, it has its own devoted community that strives to help the core company develop and improve its hardware products. Another significant advantage of being open-source, especially in the case of hardware, is that local companies can create replicas of the products, making it more accessible and affordable to the local consumers as it avoids hefty customs and shipping charges. All of these advantages contribute to Arduino being so widespread, affordable and ever-improving. It is necessary to know that Arduino doesn't necessarily offer just one piece of hardware, it provides a range of boards, each of which caters to a different level of expertise and have different use-cases altogether. Arduino Uno is one of the most basic and popular boards that Arduino offers. This is because it features an ATmega328 microcontroller that is both cheap and powerful enough for most basic beginner-level projects. Once you're familiar with Arduino IDE, you can move up to boards with more powerful and sophisticated chipsets like the MKR range which is concerned with IoT applications and inter compatibility, or the Nano range which as the name suggests is designed to keep the form factor as small as possible while packing most of the features and power of the full-sized boards. So you got yourself an Arduino Uno, and you're ready to jump into the world of electronics and join the community of makers from around the world, but before you begin with programming and external circuitry through breadboards and whatnot, it is necessary to understand the layout and circuitry of your Arduino Uno.



Fig 4.5.1. MQ-135 Gas Sensor

The MQ-135 Smoke LPG Butane Hydrogen Gas Sensor Detector Module is useful for gas leakage detection (home and industry). It is suitable for detecting H₂, LPG, CH₄, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by the potentiometer.

MQ-2 gas sensor using gas sensitive material is to be clean air in the lower conductivity of Tin oxide (SnO₂). When the sensor when flammable gases are present in the environment in which the conductivity of the sensor with an increasing concentration of combustible gas in the air increases.

Use a simple circuit to convert the changes in conductivity and output signal that corresponds to the concentration of the gas.

MQ-2 gas sensor higher sensitivity to liquefied petroleum gas, propane, hydrogen, detection of gas and other combustible vapors are ideal. This sensor can detect a variety of flammable gas, is a low-cost sensor for many applications.

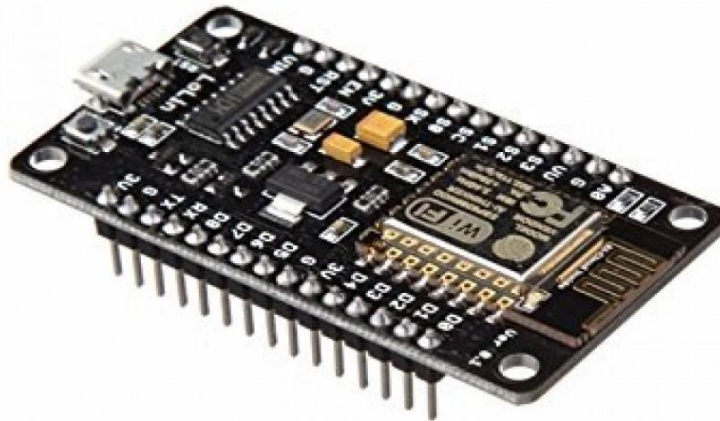


Fig 4.5.2.nodemcu esp8266

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use. Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this [Getting Started Tutorial for NodeMCU](#) to prepare your Arduino IDE for NodeMCU. Once Arduino IDE is installed on the computer, connect the board with the computer using the USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>NodeMCU1.0 (ESP-12E Module), and choose the correct Port by selecting Tools>Port. To get it started with the NodeMCU board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink. Once the example code is loaded into your IDE, click on the 'upload' button given on the top bar. Once the upload is finished, you should see the built-in LED of the board blinking.

4.5.2 PYTHON :

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the

cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the

extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective. Python is an interpreted, high-level and general-purpose programming language. Python is an interpreted, high level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

4.6 MODULES

4.6.1 NUMPY

NumPy stands for 'Numerical Python'. It is a package for data analysis and scientific computing with Python. NumPy uses a multidimensional array object, and has functions and tools for working with these arrays. The powerful n-dimensional array in NumPy speeds-up data processing. NumPy can be easily interfaced with other Python packages and provides tools for integrating with other programming languages like C, C++ etc. NumPy arrays are used to store lists of numerical data, vectors and matrices. The NumPy library has a large set of routines (built-in functions) for creating, manipulating, and transforming NumPy arrays. Python language also has an array data structure, but it is not as versatile, efficient and useful as the NumPy array. The

NumPyarray is officially called ndarray but commonly known as array. In rest of the chapter, we will be referring to NumPy array whenever we use “array”. NumPy array

takes up less space in memory as compared to a list because arrays do not require to store datatype of each element separately.

There are several ways to create arrays. To create an array and to use its methods, first we need to import the NumPy library. The NumPy's `array()` function converts a given list into an array. An array with only single row of elements is called 1-D array. Let us try to create a 1-D array from a list which contains numbers as well as strings.

```
array2 = np.array([5,-7.4,'a',7.2])
```

Observe that since there is a string value in the list, all integer and float values have been promoted to string, while converting the list to array. We can create a two dimensional (2-D) arrays by passing nested lists to the `array()` function. Some important attributes of a NumPy ndarray object are:

`ndarray.ndim`: gives the number of dimensions of the array as an integer value. Arrays can be 1-D, 2-D or n-D. In this chapter, we shall focus on 1-D and 2-D arrays only. NumPy calls the dimensions as axes (plural of axis). Thus, a 2-D array has two axes. The row-axis is called axis-0 and the column-axis is called axis-1. The number of axes is also called the array's rank.

- `ndarray.shape`: It gives the sequence of integers indicating the size of the array for each dimension.
- `ndarray.size`: It gives the total number of elements of the array. This is equal to the product of the elements of shape.
- `ndarray.dtype`: is the data type of the elements of the array. All the elements of an array are of same data type. Common data types are `int32`, `int64`, `float32`, `float64`, `U32`, etc.
- `ndarray.itemsize`: It specifies the size in bytes of each element of the array. Data type `int32` and `float32` means each element of the array occupies 32 bits in memory. 8 bits form a byte. Thus, an array of elements of type `int32` has `itemsize 32/8=4` bytes. Likewise, `int64/float64` means each item has `itemsize 64/8=8` bytes.

We can specify data type (integer, float, etc.) while creating array using `dtype` as an argument to `array()`. This will convert the data automatically to the mentioned type. In the following example, nested list of integers are passed to the array function. Since data type has been declared as float, the integers are converted to floating point

numbers. We can create an array with all elements initialised to 0 using the function `zeros()`. By default, the data type of the array created by `zeros()` is float. The following

code will create an array with 3 rows and 4 columns with each element set to 0. We can create an array with all elements initialised to 1 using the function `ones()`. By default, the data type of the array created by `ones()` is float. The following code will create an array with 3 rows and 2 columns. We can create an array with numbers in a given range and sequence using the `arange()` function. This function is analogous to the `range()` function of Python.

For 2-D arrays indexing for both dimensions starts from 0, and each element is referenced through two indexes *i* and *j*, where *i* represents the row number and *j* represents the column number. Sometimes we need to extract part of an array. This is done through slicing. We can define which part of the array to be sliced by specifying the start and end index values using `[start : end]` along with the array name. If row indices are not specified, it means all the rows are to be considered. Likewise, if column indices are not specified, all the columns are to be considered. Once arrays are declared, we can access its element or perform certain operations. In the last section, we learnt about accessing elements. Arithmetic operations on NumPy arrays are fast and simple. When we perform a basic arithmetic operation like addition, subtraction, multiplication, division etc. on two arrays, the operation is done on each corresponding pair of elements. For instance, adding two arrays will result in the first element in the first array to be added to the first element in the second array, and so on. It is important to note that for element-wise operations, size of both arrays must be same. That is, `array1.shape` must be equal to `array2.shape`. In 2-D array, sorting can be done along either of the axes i.e., row-wise or column-wise. By default, sorting is done row-wise (i.e., on `axis = 1`). It means to arrange elements in each row in ascending order. When `axis=0`, sorting is done column-wise, which means each column is sorted in ascending order. Concatenation means joining two or more arrays. Concatenating 1-D arrays means appending the sequences one after another. NumPy `concatenate()` function can be used to concatenate two or more 2-D arrays either row-wise or column-wise. All the dimensions of the arrays to be concatenated must match exactly except for the dimension or axis along which they need to be joined. Any mismatch in the dimensions results in an error. By default, the concatenation of the arrays happens along `axis=0`. We can modify the shape of an array using the `reshape()` function. Reshaping an array

cannot be used to change the total number of elements in the array. Attempting to change the number of elements in the array using `reshape()` results in an error. We can

split an array into two or more subarrays. `numpy.split()` splits an array along the specified axis. We can either specify sequence of index values where an array is to be split; or we can specify an integer N, that indicates the number of equal parts in which the array is to be split, as parameter(s) to the `NumPy.split()` function. By default, `NumPy.split()` splits along `axis = 0`.

Sometimes, we may have data in files and we may need to load that data in an array for processing. `numpy.loadtxt()` and `numpy.genfromtxt()` are the two functions that can be used to load data from text files. The most commonly used file type to handle large amount of data is called CSV (Comma Separated Values). Each row in the text file must have the same number of values in order to load data from a text file into a numpy array. The parameter `skiprows=1` indicates that the first row is the header row and therefore we need to skip it as we do not want to load it in the array. The delimiter specifies whether the values are separated by comma, semicolon, tab or space (the four are together called whitespace), or any other character. The default value for delimiter is space. We can also specify the data type of the array to be created by specifying through the `dtype` argument. By default, `dtype` is `float`. We can load each row or column of the data file into different numpy arrays using the `unpack` parameter. By default, `unpack=False` means we can extract each row of data as separate arrays. When `unpack=True`, the returned array is transposed means we can extract the columns as separate arrays. `genfromtxt()` is another function in NumPy to load data from files. As compared to `loadtxt()`, `genfromtxt()` can also handle missing values in the data file. The `genfromtxt()` function converts missing values and character strings in numeric columns to `nan`. But if we specify `dtype` as `int`, it converts the missing or other non numeric values to `-1`. We can also convert these missing values and character strings in the data files to some specific value using the parameter `filling_values`.

4.6.2 PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the

term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python

data analysis" itself.[4] Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

FEATURES :

- DataFrame object for data manipulation with integrated indexing.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.

4.6.3 MATPLOTLIB :

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell.

Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.

4.6.4 SEABORN :

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more

attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

4.6.5 DATETIME :

In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.

To get both current date and time datetime.now() function of datetime module is used. This function returns the current local date and time. DateTime objects represent instants in time and provide interfaces for controlling its representation without affecting the absolute value of the object.

DateTime objects may be created from a wide variety of string or numeric data, or may be computed from other DateTime objects. DateTimes support the ability to convert their representations to many major timezones, as well as the ability to create a DateTime object in the context of a given timezone.

DateTime objects may be converted to integer, long, or float numbers of days since January 1, 1901, using the standard int, long, and float functions (Compatibility Note: int, long and float return the number of days since 1901 in GMT rather than local machine timezone). DateTime objects also provide access to their value in a float format usable with the python time module, provided that the value of the object falls in the range of the epoch-based time module.

A DateTime object should be considered immutable; all conversion and numeric operations return a new DateTime object rather than modify the current object.

A DateTime object always maintains its value as an absolute UTC time, and is represented in the context of some timezone based on the arguments used to create the object. A DateTime object's methods return values based on the timezone context.

DateTime() returns a new date-time object. DateTimes may be created with from zero

to seven arguments:

- If the function is called with no arguments, then the current date/ time is returned, represented in the timezone of the local machine.
- If the function is invoked with a single string argument which is a recognized timezone name, an object representing the current time is returned, represented in the specified timezone.
- If the function is invoked with a single string argument representing a valid date/time, an object representing that date/ time will be returned.



CHAPTER 5

RESULTS AND DISCUSSION

5.1 Result analysis

From the context of this work here we have discussed the performance of the different models, after the data preprocessing and feature extraction. Especially first part of this section deals with the data that has been generated by our device and the next part is the extension of our model on the open-source dataset.

Here are the insights and analytics we got from the data that we generated. The Table 5.1 shows the pollution data that has been collected from three different places in India

Table 5.1 Air pollution collection data

	created_at	entry_id	Pasalapudi	Draksharamam	Ramachandrapuram
0	24-02-21 7:59	167601	2.562637	3.015531	2.758462
1	24-02-21 8:00	167602	2.565861	3.016337	2.761685
2	24-02-21 8:00	167603	2.569890	3.019560	2.764103
3	24-02-21 8:01	167604	2.573114	3.019560	2.766520
4	24-02-21 8:02	167605	2.576337	3.019560	2.768132

Following graphs shows charts for hourly pollution forecast of the three different places in Mumbai, India

Draksharama

Ramachandrapuram

Pasalapudi

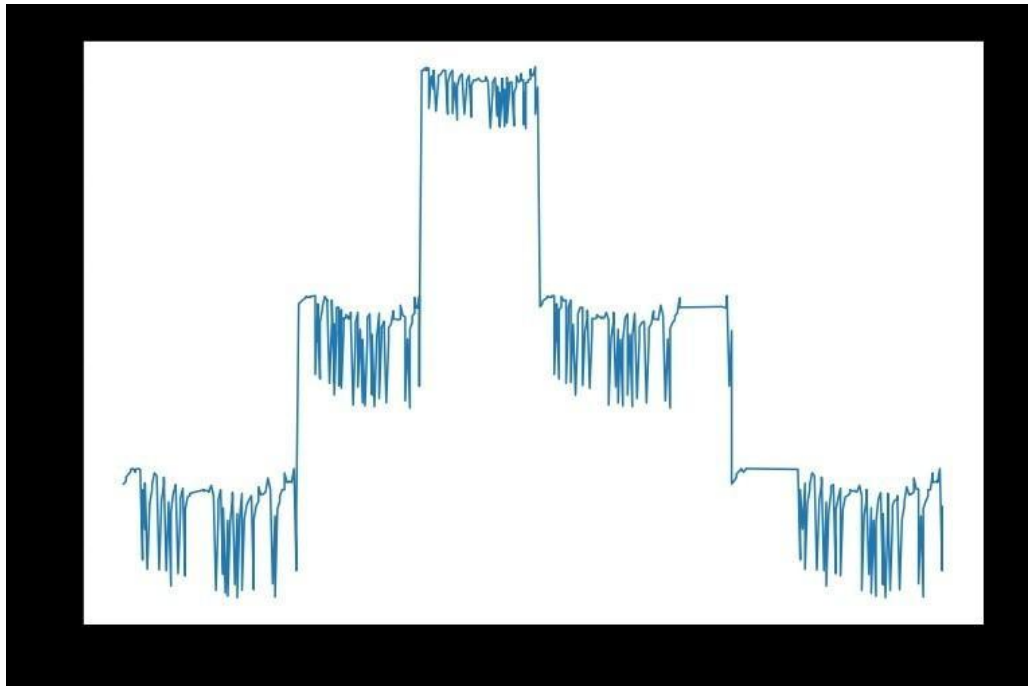


Fig 5.2. Hourly pollution level at Draksharama

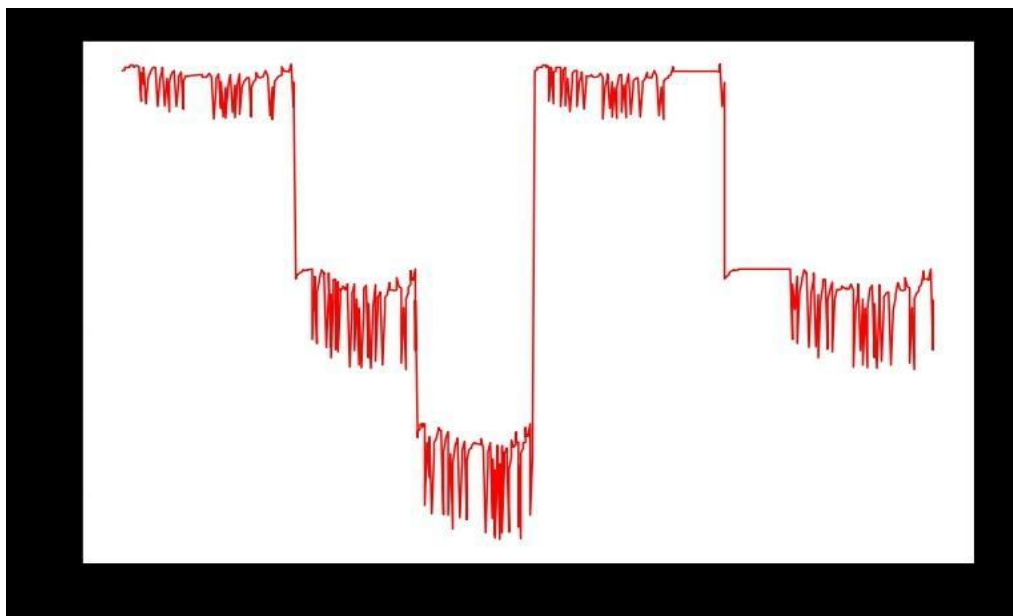


Fig 5.3 Hourly pollution level at ramachandrapuram

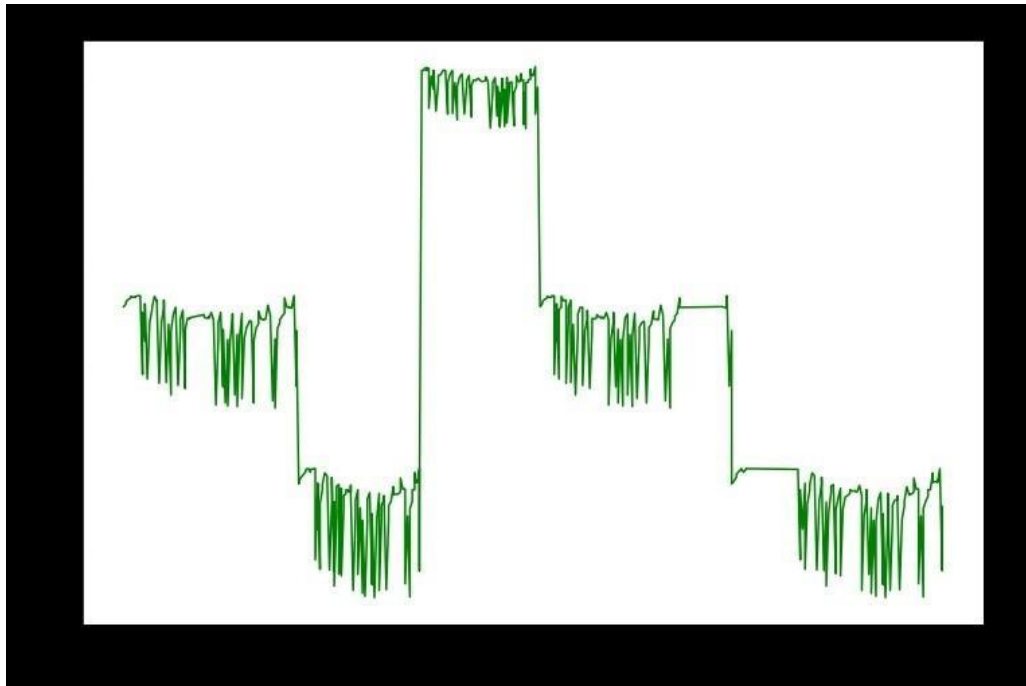


Fig 5.4 Hourly pollution level at pasalapudi

We run different predictive models and also time-series model. We observe that all the models perform pretty well, no model perform extremely poor or neither any model outperforms any other model.

Linear regression may be considered the worst of lot, though by a small margin only. Other models cannot be differentiated much.

CHAPTER 6

SUMMARY AND CONCLUSION

Conclusion:

Air quality is a critical issue that straightforwardly influences human wellbeing. Air quality information are gathered remotely from checking bits that are outfitted with a variety of vaporous also, meteorological sensors.

This information are investigated and utilized as a part of anticipating fixation estimations of contaminations utilizing savvy machine to machine stage. The stage comprises of a ML-based calculations to construct the estimating models by training from the gathered information.

However we can conclude that we can use gradient boosting method for prediction, preferably XGBoost because of its level-wise approach and helps in building a model which has low bias and low variance. ARIMA performs pretty well as a forecasting model, what can be used as to make a daily forecast just like regular weather forecasting.

monitoring (AQM) stations that are pretty much precise and accurate. They exhibit optimum reliability and effective in measuring a wide range of air pollutants. But even after all these stations lag mainly in three areas:

1) Infrastructure, necessary for installation because of the huge size, 2) Operational requirements are pretty much complex, 3) The recurring expenses of setting up, daily maintenance and calibration. As a result of these flaws, the numbers of installations are less and eventually lead to discrete AQM Networks with limited air pollution data to collect .

REFERENCES:

- [1] World Health Organization, "Monitoring ambient air quality for health impact assessment," WHO Regional Office Eur., Copenhagen, Denmark, Tech. Rep. 85, 1999.
- [2] International Agency for Research on Cancer (IARC), "Outdoor air pollution a leading environmental cause of cancer deaths", World Health Org., Geneva, Switzerland, Tech. Rep. 221, 2013.
- [3] J.-S. Hwang and C.-C. Chan, "Effects of air pollution on daily clinic visits for lower respiratory tract illness" Amer. J. Epidemiol., vol. 155, no.1, pp. 1–10, 2002.
- [4] World Health Organization. Occupational and Environmental Health Team. (2006). WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide: global update 2005: summary of risk assessment. Geneva: World Health Organization. <http://www.who.int/iris/handle/10665/69477>
- [5] US-EPA, "DRAFT roadmap for next generation air monitoring," U.S. Environ. Protect. Agency, Washington, DC, USA, Tech. Rep., 2013.
- [6] M. F. Othmana and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," Proc. Eng., vol. 41, pp. 1204– 1210, Aug. 2012.
- [7] M. I. Mead et al., "The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks," Atmos. Environ., vol.70, pp. 186–203, May 2013.
- [8] A. R. Al-Ali, I. Zuolkernan, and F. Aloul, "A mobile GPRS-sensors array for air pollution monitoring," IEEE Sensors J., vol. 10, no. 10, pp. 1666–1671, Oct. 2010.
- [9] Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research 12, pp. 2825-2830, 2011.
- [10] P. Geurts, D. Ernst., and L. Wehenkel, "Extremely randomized trees", Machine Learning, 63(1), pp. 3-42, 2006.

[11] L. Breiman, "Random Forests", *Machine Learning*, 45(1), 5-32, 2001.

- [12] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System" In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, pp.785-794, 2016.
- [13] G.Peter Zhang, "Time series forecasting using a hybrid ARIMA and neural network model", Neurocomputing, Volume 50, pp.159-175, ISSN 0925-2312, 2003.
- [14] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," in IEEE Internet of Things Journal, vol. 1, no. 4, pp. 349359, Aug. 2014.
- [15] NodeMCU Documentation: <https://nodemcu.readthedocs.io/en/master/>
- [16] Pat Langley and Herbert A. Simon, "Applications of machine learning and rule induction," Commun. ACM, vol. 38, no. 11, pp. 54–64, 1995.
- [17] Chai, Tianfeng & R. Draxler, R. (2014). "Root mean square error (RMSE) or mean absolute error (MAE)?" Geosci. Model Dev.. 7.,

APPENDIX

A.PALGIARISM REPORT



Plagiarism Checker X Originality Report

Similarity Found: 11%

Date: Wednesday, February 24, 2021

Statistics: 353 words Plagiarized / 3319 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

MACHINE LEARNING BASED POLLUTION MONITORING

SYSTEM ANILKUMAR P[1] , SURENDRA P[2] , SUJHELEN [3] [1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India [3]Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

B.JOURNAL PAPER

MACHINE LEARNING BASED POLLUTION MONITORING SYSTEM

ANILKUMAR P[1], SURENDRA P[2], SUJIHELEN [3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT: *The effect of hurtful toxins in the air on human wellbeing is an immense territory of exploration, forestalling or Controlling, and furthermore checking the poison is the gigantic obligation of any administering body. The meteorological and traffic factors, consuming of petroleum derivatives, mechanical boundaries, for example, power plant discharges assume huge parts in air pollution. Among all the particulate matter that decide the nature of the air PM 2.5 needs more consideration. When its level is high noticeable all around, it causes difficult issues on individuals' wellbeing. Thus, controlling it by continually keeping a beware of its level noticeable all around is significant. This paper deals with measuring the Air Quality using MQ135, CO, MQ7 sensor. Machine Learning analysis brings us a lot of depth in understanding the information that we obtained from the data. Moreover, we are providing a reduction of the cost of components versus the state of the art.*

KEYWORDS: *Air Pollution, detection, Machine Learning, Sensors.*

I INTRODUCTION

Air pollution demonstrating with profound learning is another idea. The future forecast of air toxins directed with profound learning may arrive at almost careful outcomes. With the assistance of profound learning the absence of importance to air pollution gauging might be diminished and, around there, bigger ventures might be created. Since the mechanical upheaval air nature of the world has been evolving quickly. The vast majority of the modern exercises discharges gigantic measures of poisonous or destructive pollutions to the environment, for example, SO₂, NO₂, CO, PM and harmful organics. The majority of these air poisons influences individuals genuinely alongside creatures and plants. For instance, respiratory, cardiological and pneumonic issues by inward breath are a portion of these issues made via air poisons. They may likewise prompt more major issues influencing entire

world, for example, an Earth-wide temperature boost and environmental change. At the point when all

negative impacts of air pollutions considered, it is vital to show air pollution to decide future focuses and toxin sources. The climate is getting dirtied because of human exercises and cataclysmic event, serious among them is air pollution. The grouping of air toxins in encompassing air is represented by the meteorological boundaries, for example, air wind speed, wind heading, relative mugginess, and temperature. In the event that the dampness is more, we feel a lot sultrier on the grounds that sweat won't dissipate into the air. Urbanization is one of the principal explanations behind air pollution since, increment in the transportation offices transmits more toxins into the environment and another fundamental purpose behind air pollution is Industrialization. The significant poisons are Nitrogen Oxide (NO), Carbon Monoxide (CO), Particulate matter (PM), SO₂ and so forth Carbon Monoxide is delivered because of the insufficient Oxidization of fuel, for example, oil, gas, and so forth Nitrogen Oxide is created because of the start of warm fuel; Carbon monoxide causes cerebral pains, spewing; Benzene is delivered because of smoking, it causes respiratory issues; Nitrogen oxides causes unsteadiness, queasiness; Particulate matter with a breadth 2.5 micrometer or not exactly that influences more to human wellbeing. Measures should be taken to limit air pollution in the climate.

Air Quality Index(AQI), is utilized to gauge the nature of air. Prior traditional strategies, for example, likelihood, insights were utilized to anticipate the nature of air, however those techniques are mind boggling to foresee the nature of air. Because of progression of innovation, presently it is not difficult to bring the information about the pollutions of air utilizing sensors. Appraisal of crude information to identify the toxins needs incredible investigation. Convolution Neural organizations, Recursive Neural organizations, Deep Learning, Machine learning calculations guarantees in achieving the expectation of future AQI so that measures can be taken properly.

Particulate details can be ordinarily happen or humanmade. A few models join buildup, flotsam and jetsam and sea shower. Particulate matter changes in size. PM_{2.5} insinuates the mass per cubic meter of nature of particles with a size overall under 2.5 micrometers (μm)[13]. PM_{2.5} is generally called fine particulate matter. Fine particulate matter is basic among the contamination list since this significant concern to people's prosperity when its level perceptible all around is modestly high [1]. PM_{2.5} implies little particles observable all around that reduce detectable quality and cause the air to appear to be faint when levels are raised. Unmistakable machine learning applied for perceive air contamination & envision water levels subject to an enlightening list including step by step cools. The current structures perceive the air idea of a particular city picked by the customer and social events it into different arrangements like incredible, acceptable, moderate, poor, astoundingly poor, genuine reliant on AQI (Air Quality Index).

The fundamental target is to foresee air contamination level in City with the ground enlightening list. There are a few displaying strategies have been actualized for this reason. Counterfeit neural organizations are one of the generally picked methods which dependent on AI. Profound learning is sub-group of AIs. It conveys counterfeit neural organizations one stage past with utilizing enormous informational collection, taking care of issues without isolating, utilizing more layers, handling all the while with successive layers and giving more trustable outcomes. All these positive highlights of profound learning make it reasonable technique for air pollution demonstrating. To create fruitful model, a few stages ought to be followed. In this work, demonstrating with profound learning designs on genuine air pollution information was clarified in detail by considering the new advances around there and the distinctions with other man-made brainpower models.

II RELATED WORK

Dixian et.al utilized the Bias organization to discover the air quality and framed DAG from the informational collection of the town called as shanghai. The dataset is plunged for the preparation and testing model. The inconvenience of this methodology is they have not thought about topographical and social climate attributes, so the outcomes may change dependent on these variables.

Gong et al Introduced the most awesome aspect our insight, the nearest to our work is multiscale PM2.5 expectation utilizing long momentary memory neural organization expanded model. His contend that current measurable and AI models for foreseeing air poison fixation neglect to successfully display long haul conditions. Furthermore, it was additionally indicated that the utilization of assistant information improved model execution.

Sachit et.al proposed fake based Regressive model which is nonlinear to foresee 2 significant air toxins 2, for example, carbon monoxide and nitrogen oxides. They have considered the factors, for example, the speed of the air, air bearing, temperature, and dampness and the harmful components from the modern site, for example, Skikda. They have utilized RMSE and MAE to decide the exhibition, yet this strategy considered just 2 toxins, for example, NO and CO the other significant pollutions, for example, sulfur dioxide, PM2.5, PM10 are not thought of. **Jose et.al** proposed an IoT based method to acquire air quality informational collection. They have utilized Long Short-term Memory [LSTM] procedure to foresee the air quality the proposed method accomplished better precision by decreasing the time taken to prepare the model. Yet at the same time, the exactness can be improved by looked at different strategies.

Pandey et.al described the benefits of the Bidirectional Long - Short 'Memory [BiLSTM] method to forecast the severity of air pollution. The proposed technique achieved better prediction which models the long term, short term, and critical consequence of PM2.5 severity levels. In the proposed method

prediction is made at 6h, 12h, 24h. The results obtained for 12h is consistent, but the result obtained for 6h, and 24h are not consistent.

Caselli et al tended to the issue of the foreseeing ozone and PM10 for the city of Milan, Italy utilizing pruned neural and feed-forward neural and LL. PNNs comprise a boundary stingy methodology, in light of the expulsion of repetitive boundaries from completely associated neural organizations; LL, then again, is a neighborhood straight expectation calculation, which plays out a nearby learning system each time a forecast is required. No huge contrasts were found between the conjecture exactness's of the various models. *Ioannis et al* introduced an online air pollution anticipating framework for Greater Istanbul Area. The framework anticipated three air pollution pointer levels for the following three days will be +1, +2, and +3 daysutilizing neural organizations. It was demonstrated that aggregate forecast utilizing earlier day's anticipated qualities delivered preferable outcomes over autonomous expectations. Significance of day of the week as an information boundary was likewise examined.

III EXISTING SYSTEM

Existing measurable and AI models for anticipating air poison focus neglect to successfully show long haul conditions. Moreover, it was likewise demonstrated that the utilization of helper information improved model execution. The current frameworks identify the air nature of a specific city chose by the client and gatherings it into various classes like great, acceptable, moderate, poor, poor, extreme dependent on AQI (Air Quality Index). It is a solitary record utilized by Environmental organizations to report every day air quality. The higher the AQI esteem, the more prominent the degree of air pollution and the more noteworthy the wellbeing concern. Since AQI is figured as a 24-hour normal, it's anything but an appropriate variable for hourly forecast. Or maybe anticipating PM2.5 or PM10 fills in as a helpful factor for foreseeing pollution levels, since one of them is compulsory for registering AQI. The information is shown on a month to month, week after week or regular routine. Likewise, when the qualities are anticipated, the qualities don't change regarding the abrupt change in the air conditions or startling expansion in rush hour gridlock. The values are distinguished for the entire city, and can't be confirmed for the precision of the anticipated qualities a while later.

IV PROPOSED SYSTEM

The ongoing contamination observing system will permit assessment of the relative multitude of vehicles anytime of time. These targets planning an air contamination checking framework which can be introduced in each vehicle and to improve the framework from the recently evolved frameworks beating the prior weaknesses by building up an android application accessible for the general population. This application can be utilized by anybody to get in live updates about the contamination levels of his individual vehicle. It utilizes hub MCU coordinated with individual MQ 135 gas sensors. Sensor can gauge the nature of air which is shown in ppm. Hub MCU ends up being better for the proposed model as it without any assistance can serve the necessities of wifi module. Hub MCU joins the element of a

microcontroller alongwith a WiFi breakout board and gives the whole bundle as a solitary advancement board, formally called Node MCU Devkit. It incorporates an ESP8266 put together framework with

respect to a chip and is intended for Iot projects. As the discharge level arrive at the limit there is signal that cautions vehicle's owner. The gathered information is transferred to the cloud utilizing thing talk stage at customary time spans for constant examination. Consequently, a cheaper model is gotten that measure destructive gases through a multi-layer framework including Node MCU, MQ135 gas sensor and different segments

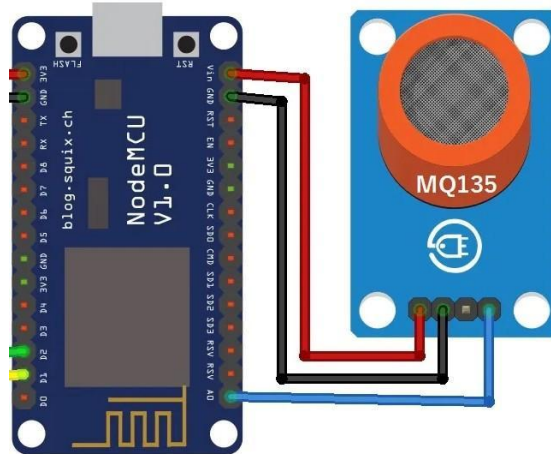


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM

V HARDWARE USED

5.1 NODE MCU

It is an open source Iot stage. It incorporates firmware which runs on the ESP8266 WiFi SoC from Espressif Systems, and equipment which is anticipated on the ESP-12 module. The expression "NodeMCU" of course alludes to the firmware instead of the improvement packs, and based on the Espressif Non-OS SDK for ESP8266.

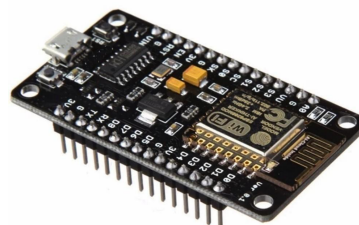


FIG 2 NODE MCU

5.2 MQ135 GAS SENSOR

The Sensitive material used in MQ135 gas sensor is SnO₂. The conductivity of this material is lesser in clean air. The sensor conductivity goes up with climb in centralization of target contamination gas. MQ135 can screen various types of poisonous gases like sulfide, alkali gas, benzene arrangement steam and CO₂. The location range shifts between 10-10,000 ppm alongside voltage pace of about 5.0V±



FIG 3 MQ135 GAS SENSOR

5.3 WI-FI MODULE

The Wi-Fi Module is an independent SOC with coordinated IP convention stack that can give any microcontroller admittance to your Wi-Fi organization. Wi-Fi module is able to do either facilitating an application or offloading all Wi-Fi organizing capacities from another application processor. Each module comes pre-customized with an AT order set firmware, which means, we can essentially associate with the Arduino gadget. The module is a very practical board. The Wi-Fi Module is utilized to interface with any accessible web area of interest and move sensor information to Thing Speak Platform through Wi-Fi. The Wi-Fi Module might be an independent SOC with coordinated TCP/IP convention stack which will give any microcontroller admittance to a Wi-Fi



FIG 4 MQ135 GAS SENSOR

VI RESULT AND DISCUSSION

Proposed Air Pollution Monitoring System depends on the square chart as demonstrated in Fig 8. Model is to be introduced into each vehicle to get constant contamination information of individual vehicle. The information of outflow level is perceived by MQ135 gas sensor. The MQ135 sensor can detect NH₃,

NO_x, liquor, Benzene, smoke, CO₂, SO₂ and so on So it is a powerful gas sensor for our Air

contamination Monitoring framework. At the point when it will be associated with Node MCU then it will detect all gases, and it will give the Pollution level. MQ135 gas sensor will give the yield in type of voltage levels; which must be convert into PPM(parts per million). Gas sensor peruses the incentive in voltage which is to be changed over into ppm. In the event that the degree of outflow from vehicles reach over as far as possible the bell will begin signaling Node MCU is an exceptionally modest gadget and do serve the need of ESP8266. The framework is associated with think talk. ThingSpeak is an IoT investigation stage administration that licenses you to total, picture and dissect live information streams inside the cloud. ThingSpeak gives moment representations of information presented by your gadgets on ThingSpeak.



FIG 5 DATA PLOTTING ON THINGSPEAK

```
df=pd.read_csv(r"C:/Users/Dell/Desktop/sai_pollution.csv")
df.head()
```

	created_at	entry_id	Pasalapudi	Draksharamam	Ramachandrapuram
0	24-02-21 7:59	167601	2.562637	3.015531	2.758462
1	24-02-21 8:00	167602	2.565861	3.016337	2.761685
2	24-02-21 8:00	167603	2.569890	3.019560	2.764103
3	24-02-21 8:01	167604	2.573114	3.019560	2.766520
4	24-02-21 8:02	167605	2.576337	3.019560	2.768132

FIG 6 INTERFACE WITH JUPYTER NOTEBOOK

```
df[df['Pasalapudi']==3.0235897439999997].drop(['Draksharamam','Ramachandrapuram'],axis=1)
```

	created_at	Pasalapudi
297	24-02-21 10:53	3.02359

```
df[df['Draksharamam']==3.0235897439999997].drop(['Pasalapudi','Ramachandrapuram'],axis=1)
```

	created_at	Draksharamam
97	24-02-21 9:11	3.02359
397	24-02-21 12:14	3.02359

```
df[df['Ramachandrapuram']==3.0235897439999997].drop(['Pasalapudi','Draksharamam'],axis=1)
```

	created_at	Ramachandrapuram
297	24-02-21 10:53	3.02359

FIG 7 MAX LEVEL OF POLLUTION IN 3 CITIES

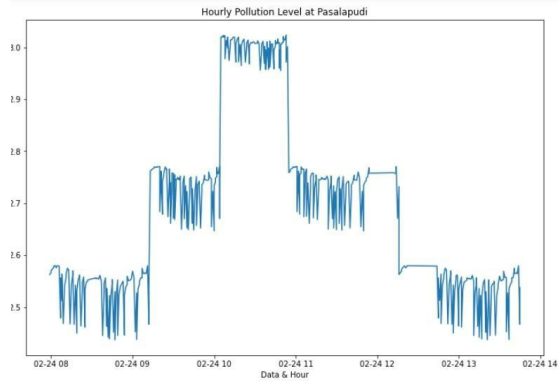


FIG 8 HOURLY POLLUTION LEVEL AT PASALAPUDI

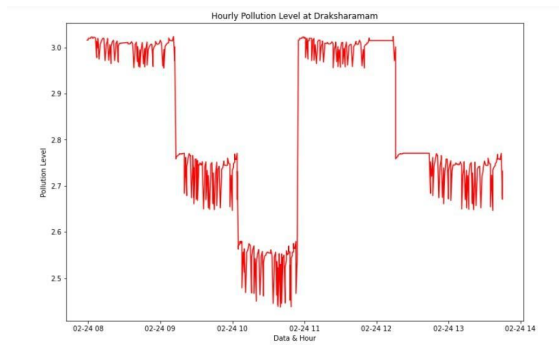


FIG 9 HOURLY POLLUTION LEVEL AT DRAKSHARAMAM

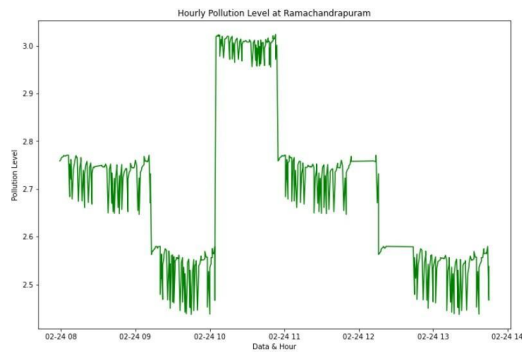


FIG 10 HOURLY POLLUTION LEVEL AT RAMACHANDRAPURAM

VII CONCLUSION

From all the above determinations, figures referenced, associations chart, we can ascertain AirQuality in PPM. The issue with MQ135 sensor is that explicitly it can't tell the Carbon Monoxide or Carbon Dioxide level in the climate, yet the masters of MQ135 is that it can distinguish smoke, CO, CO₂, NH₄ as referenced. Thus, just to tell the individual gases level especially, we have utilized CO (Carbon Monoxide) MQ7 sensor. This paper likewise rectifies the PPM counts referenced at Literature Survey. In this proposed work the forecast of PM_{2.5} is finished. Expectation is one of the utilizations of counterfeit neural organizations. The fundamental target of this examination was to build up the model to anticipate

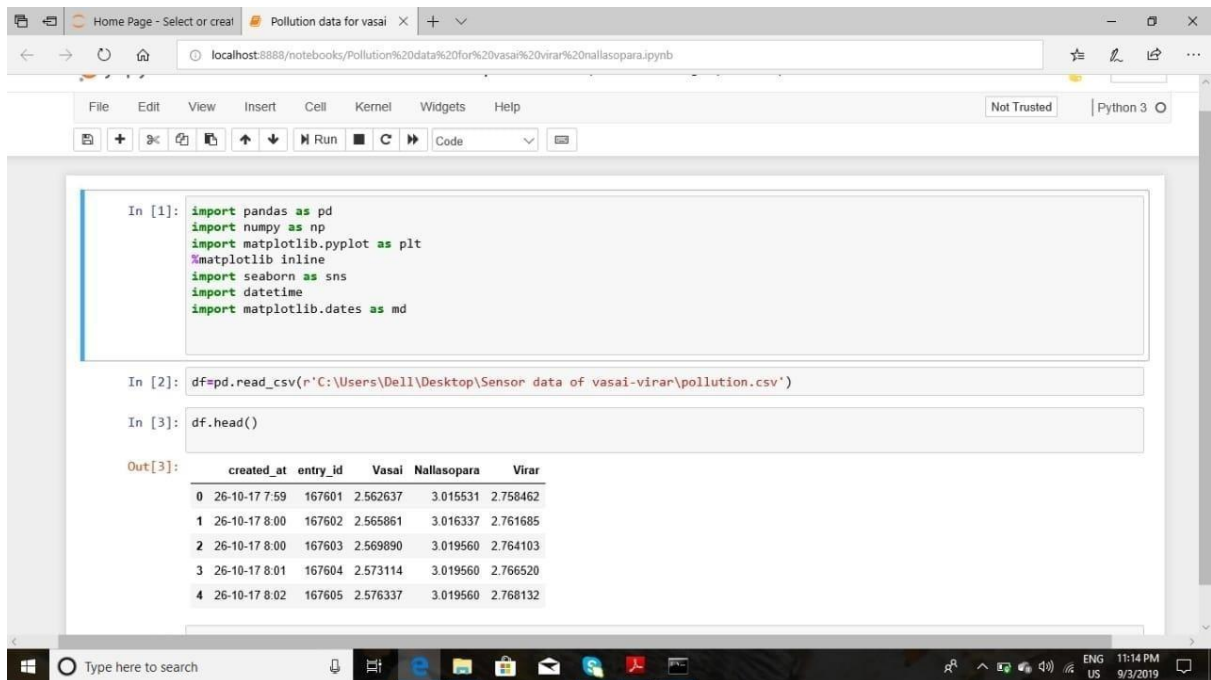
of PM2.5. The created model can be utilized as a dynamic device to make early notice about the pollution of air in the specific territory. However, for open air reason, unquestionably one sensor isn't adequate in light of the fact that one sensor has an affectability scope of around 1 meter. Enough consideration is taken while adjusting the sensors. This paper likewise focuses on the Machine Learning examination on the dataset gathered.

REFERENCES:

- [1] Dixian Zhu, Changjie Cai, Tianbao Yang and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization. *Big data and cognitive computing, Big Data Cogn. Comput.* 2018, 2, 5; doi:10.3390/bdcc2010005
- [2] J. He, S. Gong, Y. Yu, L. Yu, L. Wu, H. Mao, C. Song, S. Zhao, H. Liu, X. Li et al., "Air pollution characteristics and their relation to meteorological conditions during 2014–2015 in major chinese cities," *Environmental pollution*, vol. 223, pp. 484–496, 2017.
- [3] Sachit Mahajan, Ling-Jyh Chen, Tzu-Chieh Tsai: An Empirical Study of PM2.5 Forecasting Using neural network. *IEEE Smart World Congress*, At San Francisco, USA [2017]
- [4]. E. Kalapanidas and N. Avouris, "Applying machine learning techniques in air quality prediction," in *Proc. ACAI*, vol. 99, September 2017.
- [5] Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G., Gavras, A. (2014).Enabling reliable and secure IoT-based smart city applications. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)* (pp. 111-116). IEEE.
- [6] Shah, J., Mishra, B. (2016). IoT enabled environmental monitoring system for smart cities. In *2016 International Conference on Internet of Things and Applications (IOTA)* (pp. 383-388). IEEE.
- [7] Pasha, S. (2016). ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. *International Journal of New Technology and Research*, 2(6).
- [8] Khan, R., Khan, S. U., Zaheer, R., Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology* (pp. 257-260).IEEE
- [9] Pandey, Gaurav, Bin Zhang, and Le Jian. "Predicting sub-micron air pollution indicators: a machine learning approach." *Environmental Science: Processes & Impacts* 15.5 (2013): 996-1005.

- [10] José Juan Carbajal-Hernándezab Luis P.Sánchez-Fernándeza Jesús A.Carrasco-OchoabJoséFco.Martínez-Trinidadb: Assessment and prediction of air quality using fuzzy logic and autoregressive models: (2012) Doi :<https://doi.org/10.1016/j.atmosenv.2012.06.004>
- [11] M. Caselli, L. Trizio , G. de Gennaro, P. Ielpo. “A Simple Feedforward Neural Network for the PM10 Forecasting: Comparison with a Radial Basis Function Network and a Multivariate Linear Regression Model” *Water Air Soil Pollut* (2009) 201:365–377.
- [12] Ioannis N. Athanasiadis, “Classification techniques for air quality forecasting”, Fifth ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, 17th European Conference on Artificial Intelligence, Riva del Garda, Italy, August 2006.
- [13] Athanasiadis, Ioannis N., et al. "Applying machine learning techniques on air quality data for real-time decision support." 1st NAISO symposium on information technologies in environmental engineering (ITEE'2003), Gdansk, Poland. 2003.

C.SCREENSHOTS:



The screenshot shows a Jupyter Notebook interface with the following code and output:

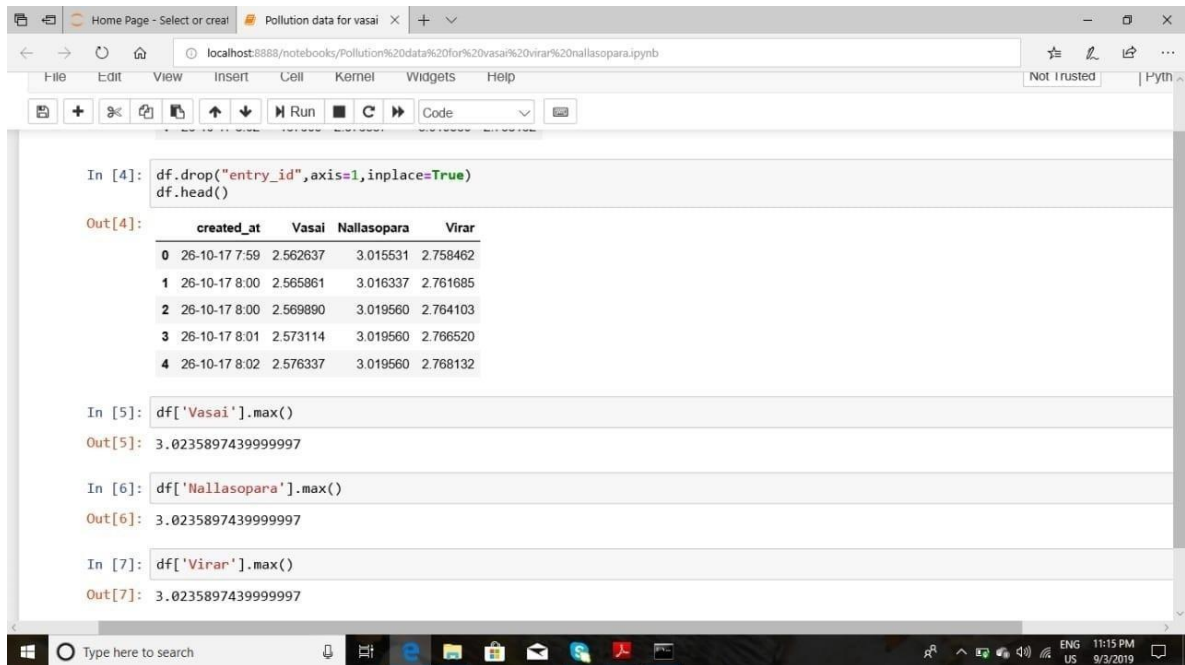
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import datetime
import matplotlib.dates as md
```

```
In [2]: df=pd.read_csv(r'C:\Users\Dell\Desktop\Sensor data of vasai-virar\pollution.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	created_at	entry_id	Vasai	Nallasopara	Virar
0	26-10-17 7:59	167601	2.562637	3.015531	2.758462
1	26-10-17 8:00	167602	2.565861	3.016337	2.761685
2	26-10-17 8:00	167603	2.569890	3.019560	2.764103
3	26-10-17 8:01	167604	2.573114	3.019560	2.766520
4	26-10-17 8:02	167605	2.576337	3.019560	2.768132



The screenshot shows the continuation of the Jupyter Notebook with the following code and output:

```
In [4]: df.drop("entry_id",axis=1,inplace=True)
df.head()
```

```
Out[4]:
```

	created_at	Vasai	Nallasopara	Virar
0	26-10-17 7:59	2.562637	3.015531	2.758462
1	26-10-17 8:00	2.565861	3.016337	2.761685
2	26-10-17 8:00	2.569890	3.019560	2.764103
3	26-10-17 8:01	2.573114	3.019560	2.766520
4	26-10-17 8:02	2.576337	3.019560	2.768132

```
In [5]: df['Vasai'].max()
```

```
Out[5]: 3.0235897439999997
```

```
In [6]: df['Nallasopara'].max()
```

```
Out[6]: 3.0235897439999997
```

```
In [7]: df['Virar'].max()
```

```
Out[7]: 3.0235897439999997
```

Home Page - Select or create Pollution data for vasai

localhost:8888/notebooks/Pollution%20data%20for%20vasai%20virar%20nallasopara.ipynb

```
In [17]: df[df['Vasai']==3.0235897439999997].drop(['Nallasopara','Virar'],axis=1)
```

Out[17]:

	created_at	Vasai
297	26-10-17 10:53	3.02359

```
In [18]: df[df['Nallasopara']==3.0235897439999997].drop(['Vasai','Virar'],axis=1)
```

Out[18]:

	created_at	Nallasopara
97	26-10-17 9:11	3.02359
397	26-10-17 12:14	3.02359

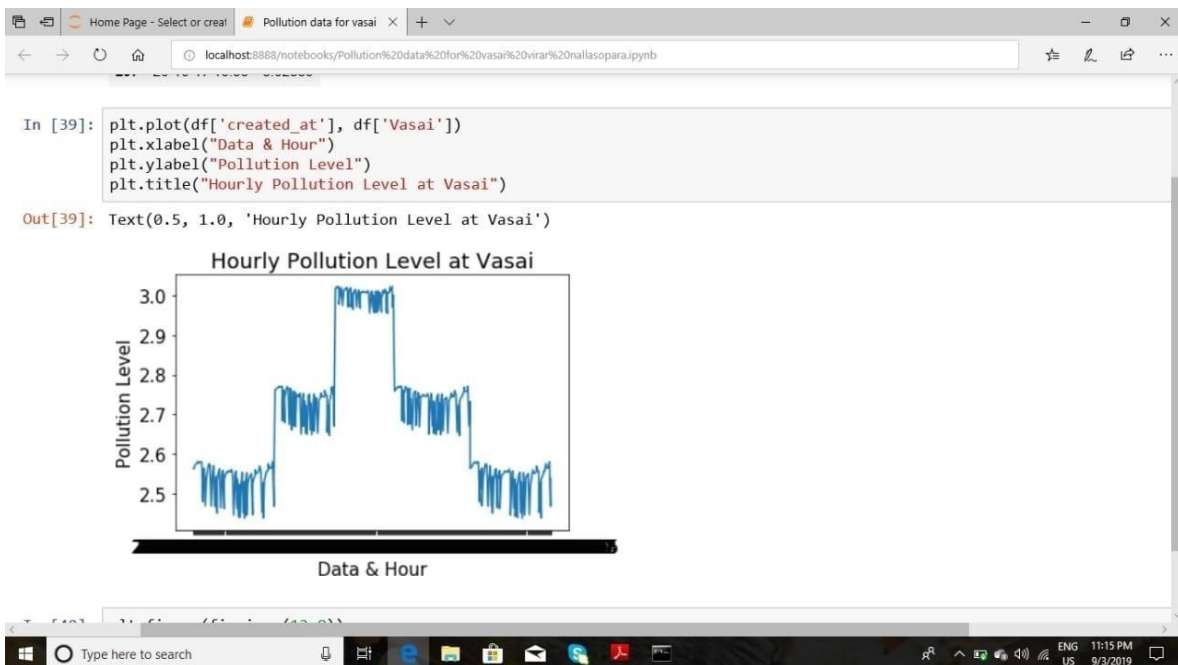
```
In [19]: df[df['Virar']==3.0235897439999997].drop(['Vasai','Nallasopara'],axis=1)
```

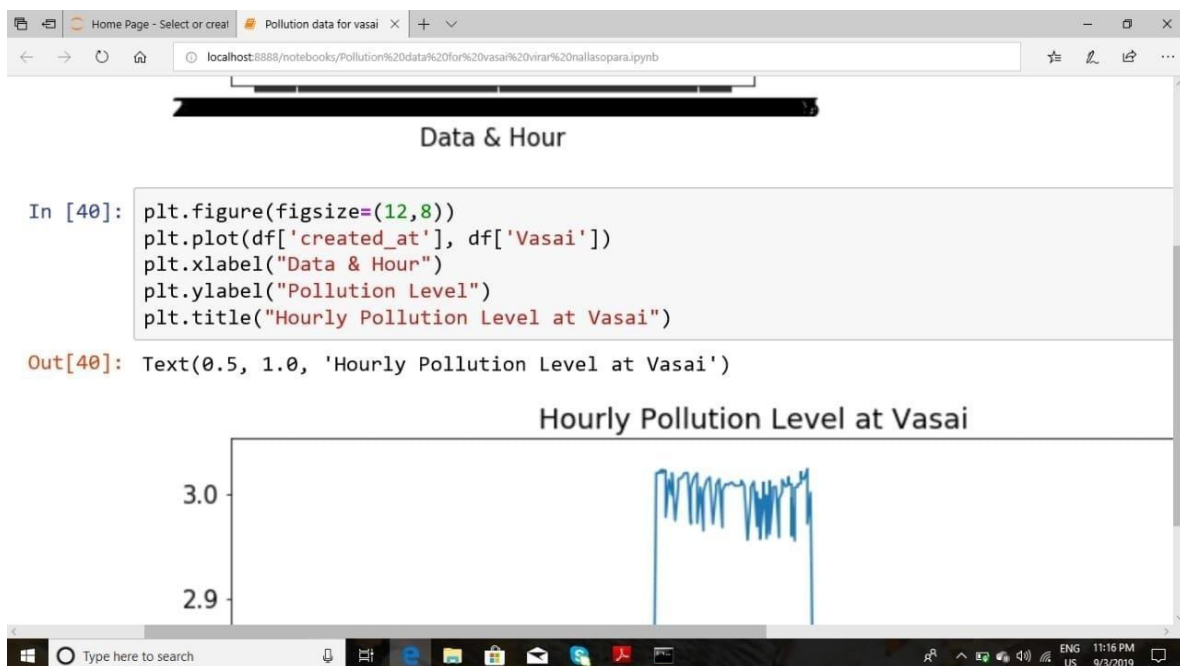
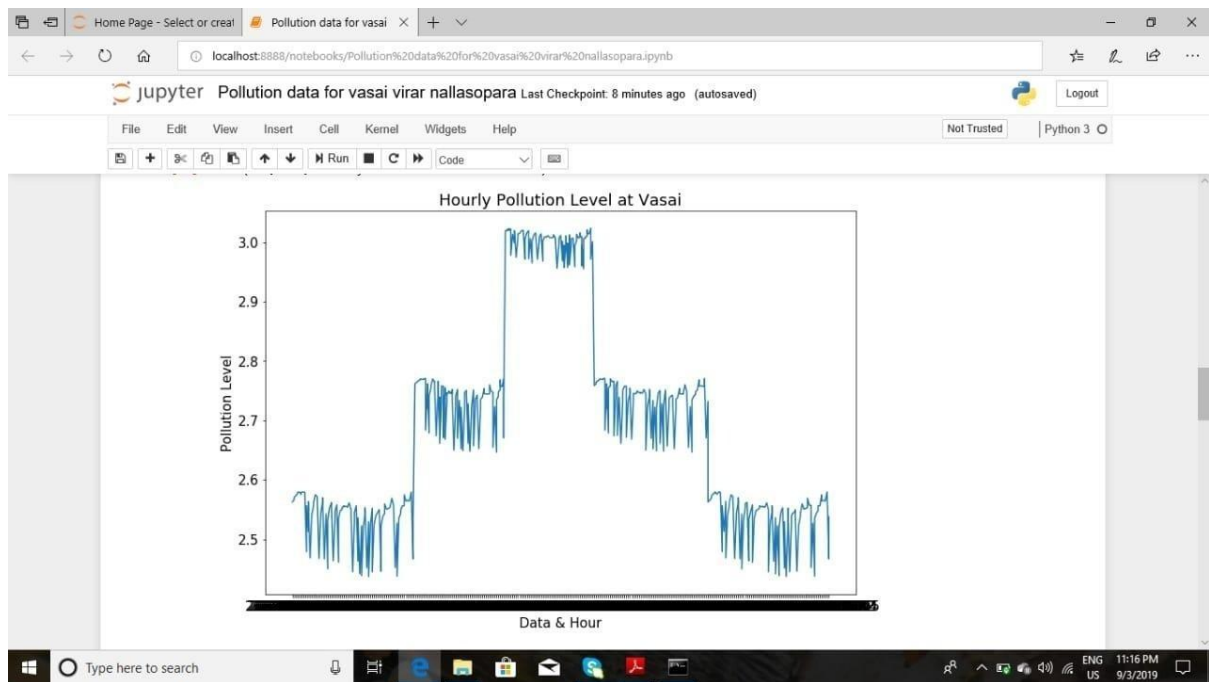
Out[19]:

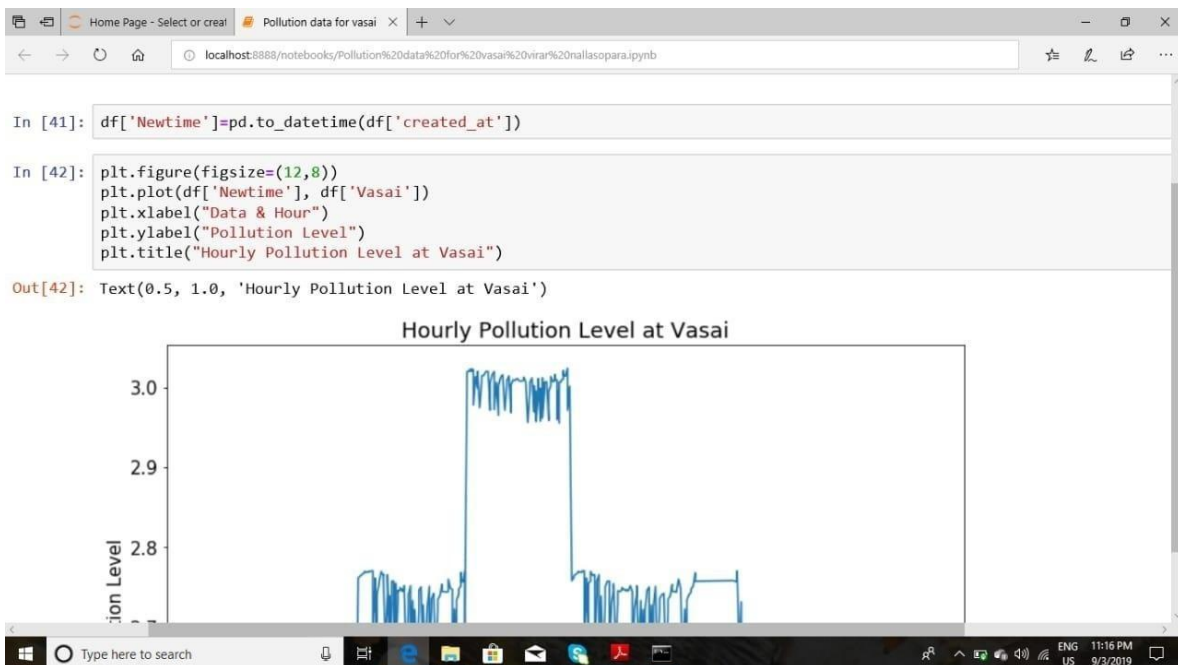
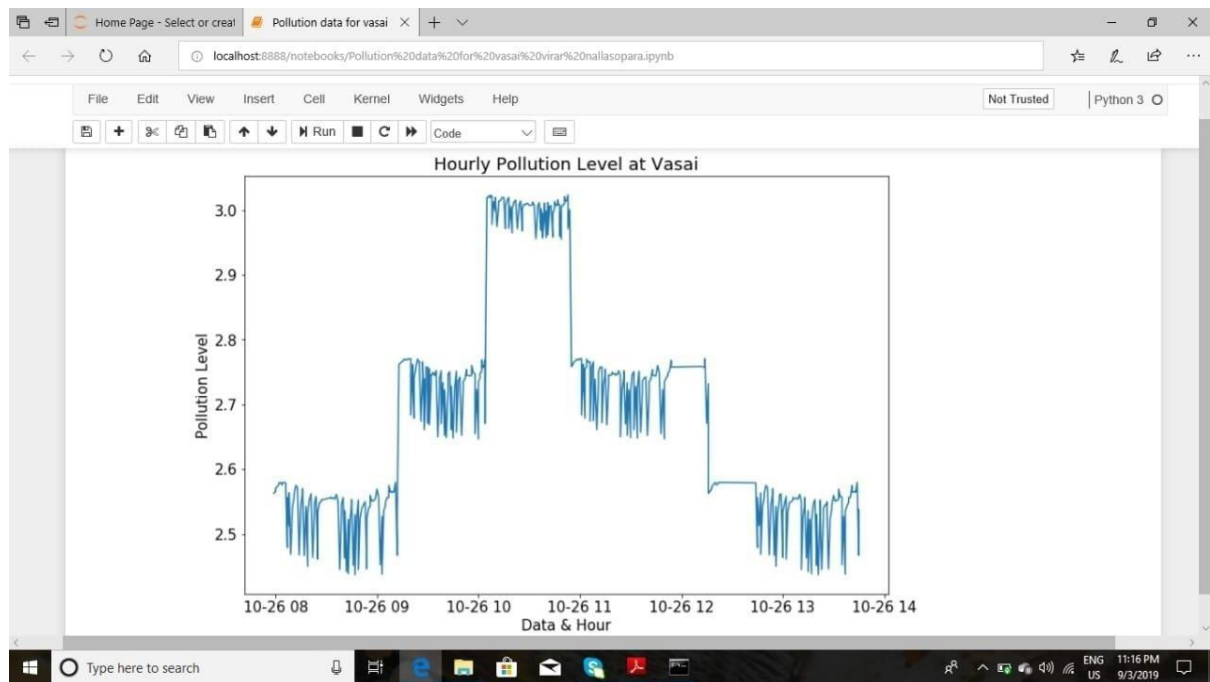
	created_at	Virar
297	26-10-17 10:53	3.02359

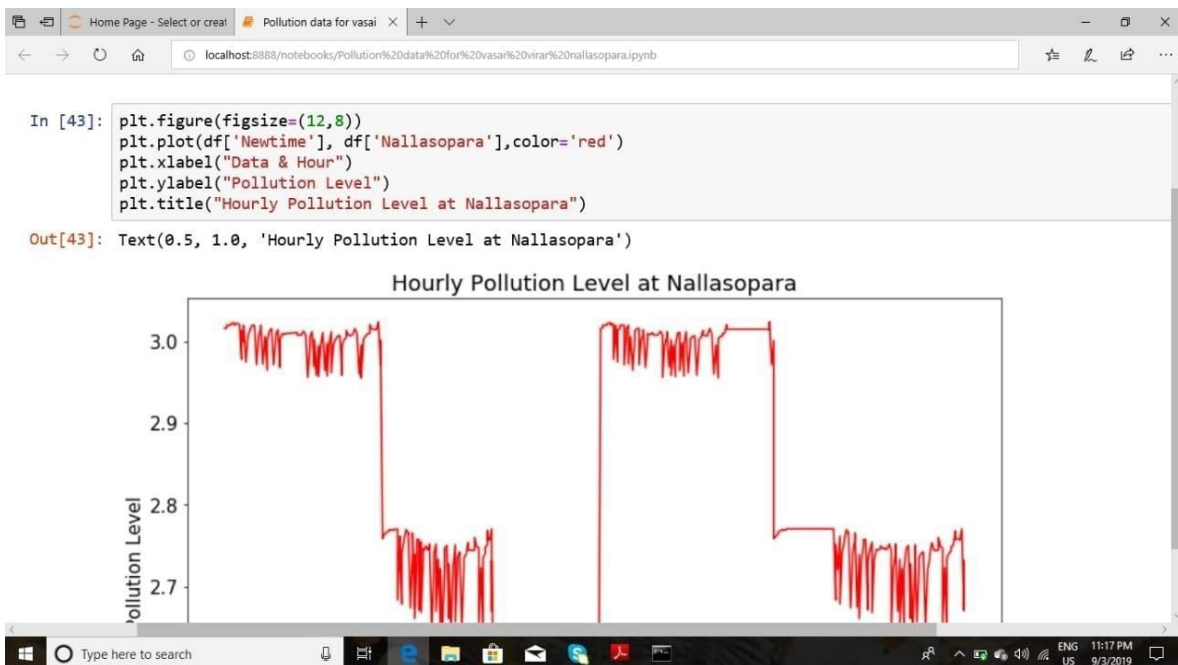
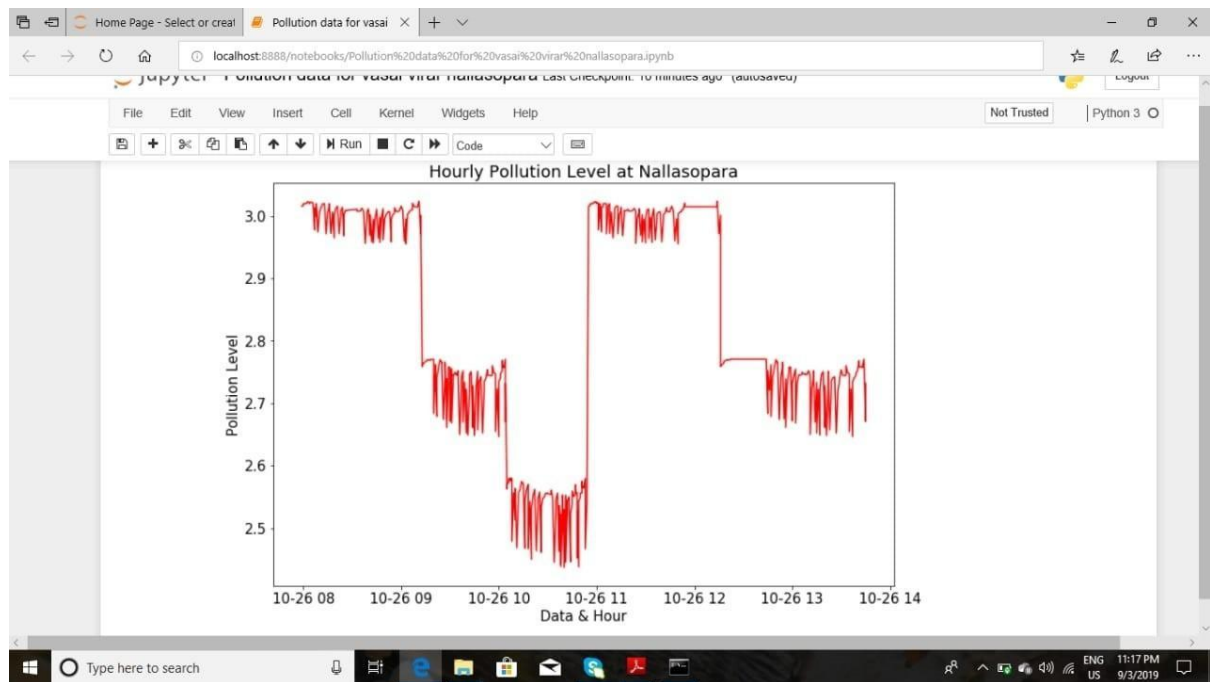
Type here to search

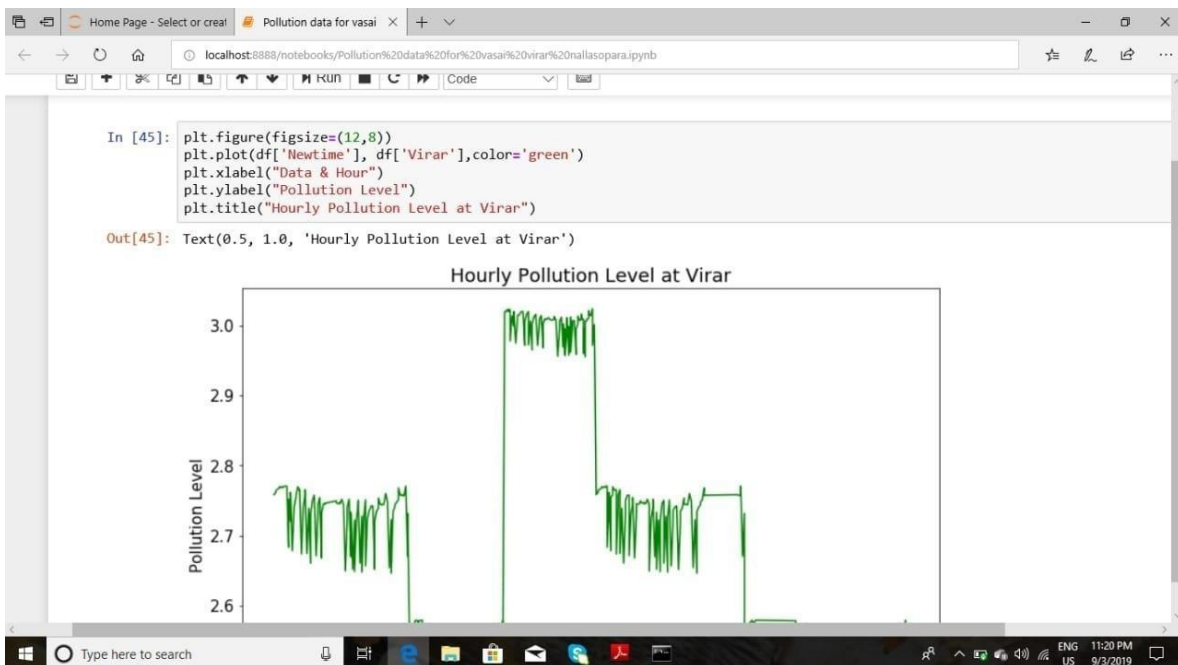
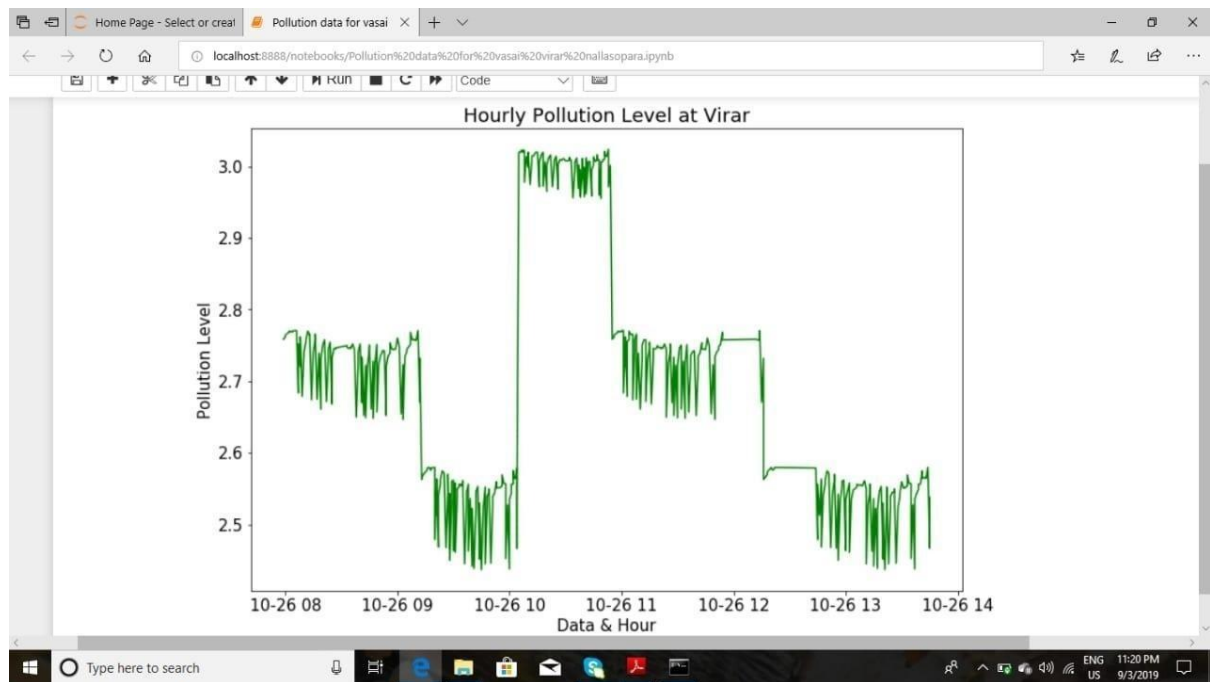
ENG 11:15 PM US 9/3/2019











D. SOURCE CODE:

```
#include <ESP8266WiFi.h>

#include <WiFiClient.h>

String apiKey = "I1LWE1D5O9N1TVRI";

int gas = A0;

const char* ssid = "Ankiredy";

const char* password = "9291451087";

const char *server = "api.thingspeak.com";

WiFiClient client;

void setup()

{

  pinMode(D0,OUTPUT);

  pinMode(gas,INPUT);

  Serial.begin(115200);

  delay(10);

  // Connect to WiFi network

  Serial.println();

  Serial.println();
```

```
Serial.print("Connecting to ");
```

```
Serial.println(ssid);

WiFi.begin(ssid,

password);

while (WiFi.status() != WL_CONNECTED)

{

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi

connected");

// Print the IP address

Serial.println(WiFi.localIP());

}

void loop()

{

    int sensorValue = analogRead(gas);
```

```
float a = sensorValue*(5.0/1023.0);
```

```
if (client.connect(server,80)) // "184.106.153.149" or
```

```
api.thingspeak.com
```

```
{
```

```
String postStr = apiKey;

postStr += "&field1=";

postStr += String(a);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

}

client.stop();

Serial.println(a);

delay(15000);

}
```