JSRT JOURNAL

PAGES: 01-15
02/09/25

JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY VOLUME-3 ISSUE-9 SEPTEMBER

Air Quality Prediction Using Machine Learning Techniques: A Case Study On Sulphur Dioxide Forecasting

Sunil[1], Dr. Shilpa B Kodli[2]

[1] *MTech Student, Department of Computer Science and Engineering, VTU CPGS, Kalaburagi, India,* vsunilpandit@gmail.com

[2]*Prof, Department of Computer Science and Engineering, VTU CPGS,Kalaburagi, India.* shilpakodli@gmail.com

ABSTRACT

Air pollution is one of the greatest problems being faced by mankind. Millions of people die each year because of reasons directly or indirectly related to air pollution. Effective strategies to counter the harmful effects of air pollution are an imperative need of the times. The responses to the air pollution problems are usually knee-jerk reactions, which don't help in the long run. For developing an effective counter-strategy for combating air pollution, it is necessary to focus the efforts on the pollutants that are most responsible for the air pollution. Examining and protecting air quality has become one of the most essential activities for the government in many industrial and urban areas today. The meteorological and traffic factors, burning of fossil fuels, and industrial parameters play significant roles in air pollution. With this increasing air pollution, We are in need of implementing models which will record information about concentrations of air pollutants(so2,no2,etc).The deposition of this harmful gases in the air is affecting the quality of people's lives, especially in urban areas. Lately, many researchers began to use Big Data Analytics approach as there are environmental sensing networks and sensor data available. In this project, machine learning techniques are used to predict the concentration of so2 in the environment. Sulphur dioxide irritates the skin and mucous membranes of the eyes, nose, throat, and lungs. Models in time series are employed to predict the so2 readings in nearing years or months.

Keywords – Air quality, prediction, Machine learning.

I.INTRODUCTION

Air pollution is one of the biggest problems that the world is facing today. Although the last year has seen some improvements in quality of air because of the widespread lockdowns and reduced activity levels because of Covid-19, but as the world population is starting to reassume normalcy, the levels of air pollution have again started to reach alarming levels. According to the reports published by World Health organization and State of Global Air, 21 of the 30 most polluted cities in the world are in India and more alarmingly air pollution contributes to death of about 2 million people in India annually. The different states of India face varying levels of pollution throughout the year based on socio economic, agricultural, industrial and other factors. The levels of pollution are usually measured in terms of Air Quality Index. Different countries have their own indices and their specific method of calculating the indices, but all of them involve measurement of the relative concentrations of the pollutants and using these concentration values to evaluate the index.

Figure 1 shows the various factors that contribute to the air pollution in India as well as the various pollutants that are considered for calculation of Air Quality Index(AQI) in India. The concentrations of the pollutants are recorded from the surface- mounted sensors that are deployed at various locations in cities all over the country. the data is then used to calculate AQI which is then used to indicate the quality of air or the level of air pollution. The Indian AQI is specified by a number between 0 and 500 and is then categorized using one of the six levels: good, satisfactory, moderate, poor, very poor and severe depending on the concentrations of the various pollutants and the effect of air quality on health.

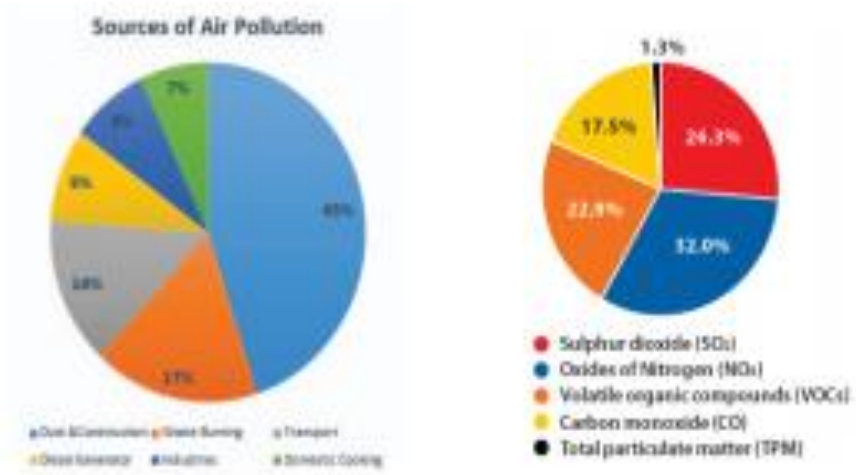


Fig 1: Sources of Pollution and Major Pollutants

In the developing countries like India, the rapid increase in population and economic upswing in cities have lead to environmental problems such as air pollution, water pollution, noise pollution and many more. Air pollution has direct impact on humans health .There has been increased public awareness about the same in our country. Global warming, acid rains, increase in the number of asthma patients are some of the long-term consequences of air pollution. Precised air quality forecasting can reduce the effect of maximal pollution on the humans and biosphere as well. Hence, enhancing air quality forecasting is one of the prime targets for the society. Sulphur Dioxide is a gas. It is one of the major pollutants present in air. It is colorless and has a nasty, sharp smell. It combines easily with other chemicals to form harmful substances like sulphuric acid, sulfurous acid etc. Sulfur dioxide affects human health when it is breathed in. It irritates the nose, throat, and airways to cause **coughing, wheezing, shortness of breath**, or a tight feeling around the chest. The concentration of sulphur dioxide in the atmosphere can influence the **habitat suitability** for plant communities, as well as animal life.
The proposed system is capable of predicting concentration of Sulphur Dioxide for forthcoming months / years.

1.1 Problem Statement

Air pollution is one of the greatest problems being faced by mankind. Millions of people die each year because of reasons directly or indirectly related to air pollution. Effective strategies to counter the harmful effects of air pollution are an imperative need of the times. The responses to the air pollution problems are usually knee-jerk reactions, which don't help in the long run. For developing an effective counter-strategy for combating air pollution, it is necessary to focus the efforts on the pollutants that are most responsible for the air pollution.

With increasing public health consciousness, many cities have established air quality monitoring locations. However, most services only show the current air quality and do not forecast air quality.

Air quality prediction is essential to help guiding individual actions limiting PM2.5 exposure, e.g., choosing outdoor or indoor activities. However, accurate air quality forecasting is hindered by a complex array of factors, including emissions, traffic patterns, and meteorological conditions.

1.2 Objectives

The main objective is focus on identifying the pollutant that plays the most important role in defining the Air Quality Index of a region and also attempts to establish the trend patterns followed by the identified pollutant. This will help the decision makers to devise counter-strategies well in advance for countering the harmful effects of the main pollutant thereby helping to reduce air pollution.

II.LITERATURE SURVEY & EXISTING SYSTEM

2.1 Literature Survey:

Detection and Prediction of Air Pollution using Machine Learning Models

In this paper, Logistic regression is employed to detect whether a data sample is either polluted or not polluted and Autoregression is employed to predict future values of PM2.5 based on the previous PM2.5 readings. Knowledge of level of PM2.5 in nearing years, month or week, enables us to reduce its level to lesser than the harmful range. This system attempts to predict PM2.5 level and detect air quality based on a data set consisting of daily atmospheric conditions in a specific city. The dataset used in this system has the following attributes - temperature, wind speed, dewpoint, pressure, PM2.5 Concentration(ug/m^3) and the classification result – data sample is classified as either polluted or not polluted. Based on the logit function, the Logistic Regression model classifies the training data to be either 0 (not polluted) or 1 (polluted) and accuracy is verified using the test data. The Autoregressive model modifies the dataset into time series dataset by taking the date and previous PM2.5 values from the main data set and makes the future predictions.

Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms

In this paper, support vector regression (SVR) and random forest regression (RFR) are used to build regression models for predicting the Air Quality Index (AQI) in Beijing and the nitrogen oxides (NOX) concentration in an Italian city, based on two publicly available datasets. In this experiment, the AQI of Beijing is taken as the regression target. For the SVR-based model training, radial basis function (RBF) is chosen as the kernel function. The kernel parameter gamma (γ) and the penalty parameter (C) are selected by a grid search method. For the RF-based model, 100 regression trees were used to build the regression model. The rootmean-square error (RMSE), correlation coefficient (r), and coefficient of determination (R2) were used to evaluate the performance of the regression models. This work also illustrates that combining machine learning with air quality prediction is an efficient and convenient way to solve some related environment problems.

2.2 Existing System

An integrated model using Naïve Bayes and ANN to predict the level of air pollutants at various locations in Mumbai and Navi Mumbai using past data available from meteorological department and Pollution Control Board. The proposed model is implemented and tested using MATLAB for ANN an R for Kriging and the results are presented.

This system has used the Naïve Bayes and Multilayer Perceptron (ANN) Protocol for prediction of the pollution of next day. The system helps to predict next date pollution details based on basic parameters and analyzing pollution details and forecast future pollution.

III.PROPOSED SYSTEM & METHODOLOGY

3.1 Proposed System:

Linear regression and Random Forest is used to detect whether a data sample is either polluted or not polluted and it will show more accuracy compared to existing methods. And machine learning techniques for air quality evaluation and prediction.

Advantages

The proposed classification algorithms show More Accuracy.

3.2 Methodology:

We do **Data collection, Data Preprocessing, Classification and Performance measures** as modules.

Data Collection:

I collect the data from the https://en.tutiempo.net/climate/ws-431280.html from 2013 to 2018 using requests module.

we have (Input attributes) T =Average Temperature (°C) ,TM = Maximum temperature (°C) ,Tm == Minimum temperature (°C) , SLP = Atmospheric pressure at sea level (hPa) , H = Average relative humidity (%) ,VV = Average visibility (Km) ,V = Average wind speed (Km/h) ,VM =Maximum sustained wind speed (Km/h).

We also need PM 2.5 (Output attribute) value, PM 2.5 values are collected from paid API from 2013 to 2018.

Data Preprocessing:

In this stage, we have to do data removal, data transformation, there we remove the fields unwanted and we modify or replace the values in the null fields.

IV.SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

- System : Intel IV 2.4 GHz.
- Hard Disk : 100 GB.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- RAM :4/8 GB.

4.2 SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

- Operating system : Windows XP/7/10/11
- Coding Language : Python
- Develpoment Kit : VS Code/ Spyder/ Pycharm
- Programming Language : Python
- IDE : Anaconda

V.FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Economical Feasibility
- Operation Feasibility

5.1 TECHNICAL FEASIBILITY:

In the feasibility study first step is that the organization or company has to decide that what technologies are suitable to develop by considering existing system.

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to hold the data required to use the new system?

- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Here in this application used the technologies like **Visual Studio 2012 and SqlServer** 2014. These are free software that would be downloaded from web.

Visual Studio 2013 –it is tool or technology.

5.2 ECONOMICAL FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

5.3 OPERATIONAL FEASIBILITY:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following

- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

management issues and user requirements have been taken into consideration. So there is no This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

VI. IMPLEMENTATION

6.1 Software Development Life Cycle:

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development.

Fig 2: SDLC Cycle

Requirements:
Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.

Design:
The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

Implementation:
Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

Testing:
During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model. Now let’s delve into some of the traditional and widely used variations.

6.2 SDLC METHDOLOGIES:
This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.
As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The following diagram shows how a spiral model acts like:

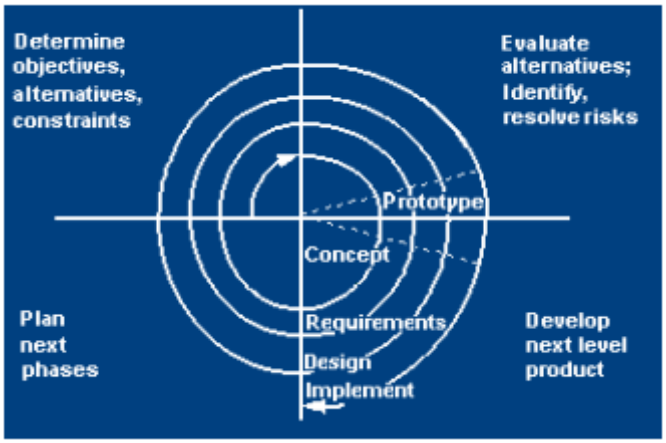


Fig 3: Spiral Model

The steps for Spiral Model can be generalized as follows:
• The new system requirements are defined in as much details as possible. This usually involves interviewing a number of usersrepresenting all the external or internal users and other aspects of the existing system.
• A preliminary design is created for the new system.
• A first prototype of the new system is constructed from the preliminary design.
This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
• A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning a designing the second prototype.
4. Constructing and testing the second prototype.

• At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
• The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
• The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
• The final system is constructed, based on the refined prototype.
• The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimizendown time.

6.3 STUDY OF THE SYSTEM:
In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows
1. Administrative User Interface Design
2. The Operational and Generic User Interface Design
The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

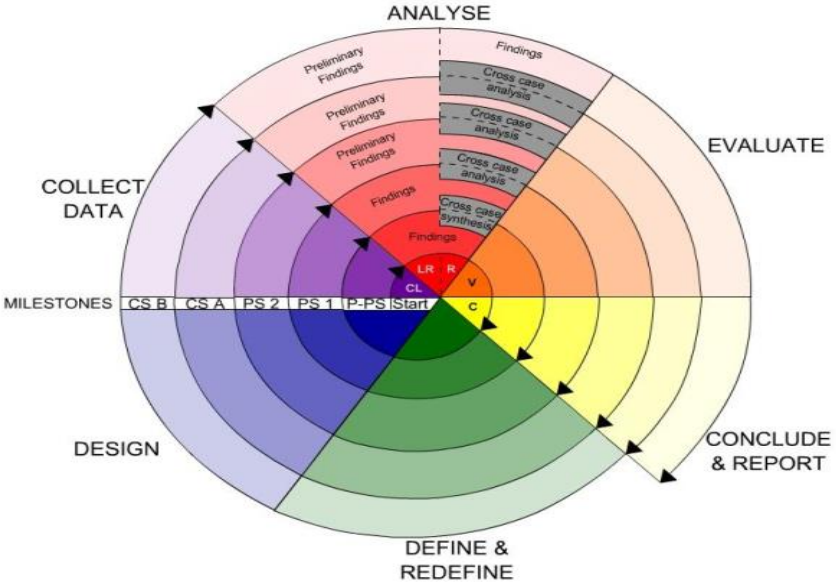


Fig 4: Operational user Interface

VII.TESTING

Testing is really a progression of various tests whose main role is to completely practice the PC based framework. Albeit every test has an alternate reason, all work to check that all the framework components have been legitimately coordinated and perform apportioned capacities .The testing procedure is really completed to ensure that the item precisely does likewise what should do. Testing is the last check and acceptance action inside of the association itself.

In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code.

7.1 Unit Testing

Here each module that comprises the overall system is tested individually. Unit testing focuses verification efforts even in the smallest unit of software design in each module. The modules of the framework are tried independently. This testing is done in the programming style itself. Unit testing practices particular ways in a module's control structure to guarantee complete scope and most extreme blunder location.

7.2 Integration

After successful completion of unit testing or module testing, individual functions are integrated into classes. Again integration of different classes takes into place and finally integration of front-end with back-end occurs.

- **Integration of functions into classes**

At the start of coding phase only the functions required in different parts of the program are developed. Each of the functions is coded and tested independently. After verification of correctness of the different functions, they are integrated into their respective classes.

- **Integration of different classes**

  Here the different classes are tested independently for their functionality. After verification of correctness of outputs after testing each class, they are integrated together and tested again.

- **Integration of front-end with back-end**

  The front-end of the project is developed in Python Swing environment. The user interface is designed to facilitate the user to input various commands to the system and view the system's normal and faulty behavior and its outputs. The back-end code is then integrated with the GUI and tested.

7.3 Integration Testing

Integration testing is a systematic technique for constructing the program structure. It addresses the issues associated with the dual problems of verification and program construction. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.

After the software has been integrated, a set of high order tests are conducted. All the modules are combined and tested as a whole. Here correction is difficult, because the isolation of errors is complicated by the vast expanse of the entire program.

7.3.1 Top down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward, beginning with the main program module.

7.3.2. Bottom-up Integration

. The following integration testing table shows the functions that were combined into different classes and the class as a whole tested for its functionality. This is important to check for error free interaction between various classes, and maintenance of data integrity.

7.3.3. Bottom-up Integration

. The following integration testing table shows the functions that were combined into different classes and the class as a whole tested for its functionality. This is important to check for error free interaction between various classes, and maintenance of data integrity.

Classes integrated	Functions integrated in each class	Tests done	Remarks
Class: Main	UploadDataset() ExtractFeatures() Train() Predict()	Class tested to check whether all operations that were applied are working correctly.	Success
Class: Training	extractDatset() selectFeatures() preprocess() Train()	Class tested to check whether all operations that were applied are working correctly.	Success

Class: Predictive Models	predictBasedonRandomForest() predictBasedOnLinearRegression() predictBasedOnSVM() predictBasedOnNeuralNetwork()	Class tested to check whether all operations were applied are working correctly and appropriately or not.	Success

7.4 Validation Testing

At the finish of combination testing, programming is finished and gathered as a bundle. Interfacing blunders are revealed and remedied. Approval testing can be characterized from multiple points of view. Here the testing approves the product capacity in a way that is sensibly expected by the client

Table 1 :- Validation testing table

7.5 Output Testing

Functionality to be tested	**Input**	**Tests done**	**Remarks**
Working of Front End	Appropriate forms open when User Clicks.	Forms Validated	Success
Working of Preprocessing	Working of Dataset loaded and preprocessed successfully.	Preprocess done.	Success
Working Classification and performance computation	Check for prediction with Appropriate algorithms and view graph.	Predicted result should display and Graph should display accuracy.	Success

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Therefore the output testing involves first of all asking the users about the format required by them and then to test the output generated or displayed by the system under consideration. The output format is considered in 2 ways: –

- On screen
- Printed format

7.6 User Acceptance Testing

User Acceptance of a system is the key factor to the success of any system. Performance of an acceptance test is actually the user's show. User motivation and knowledge are critical for the successful performance of the system.

The system under consideration is tested for user acceptance by constantly in touch with the prospective system users at time of developing and making changes wherever required in regard to the following point:

- Input Screen design
- Output Screen design
- Menu driven system

7.6.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit.

Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity

7.6.2Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:-

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.
- Errors in objects.

Advantages

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

7.7 Preparation of Test Data

Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by using test data, errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

7.7.1 Using Artificial Test Data

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

7.8 Quality Assurance

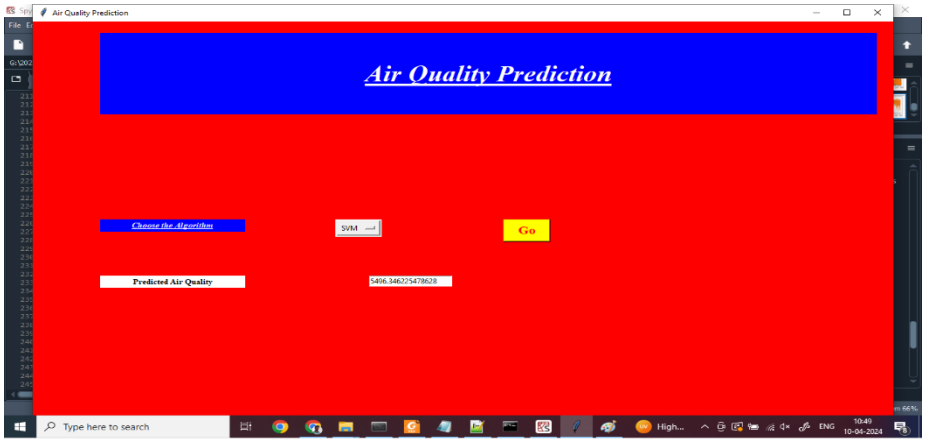
Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an “umbrella activity” that is applied throughout the engineering process. Software quality assurance encompasses:-

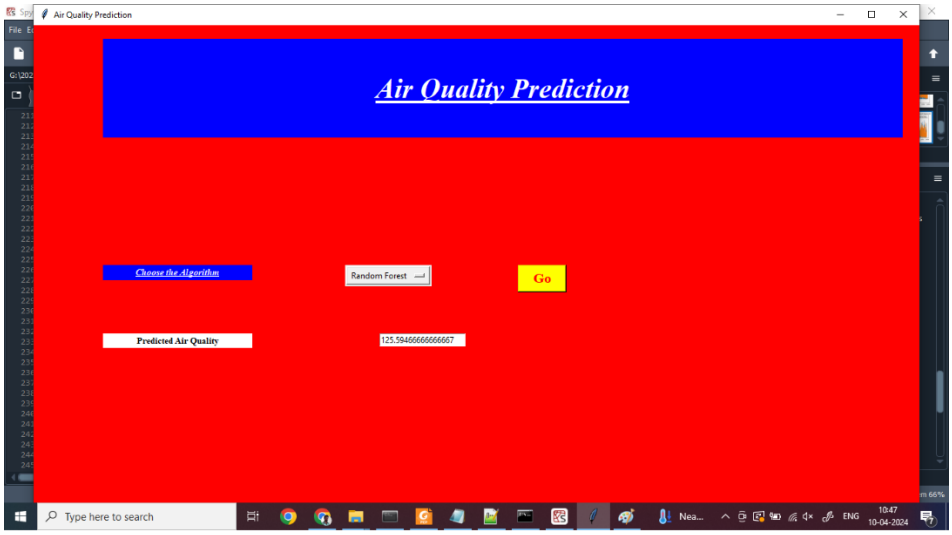
- Analysis, design, coding and testing methods and tools
- Formal technical reviews that are applied during each software engineering
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

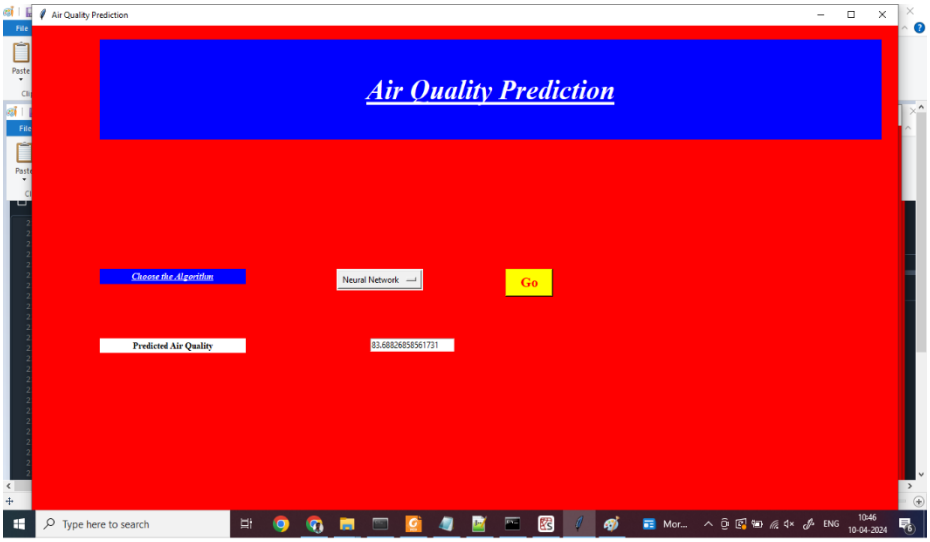
VIII. INTERPRETATION OF RESULT or SCREENSHOTS

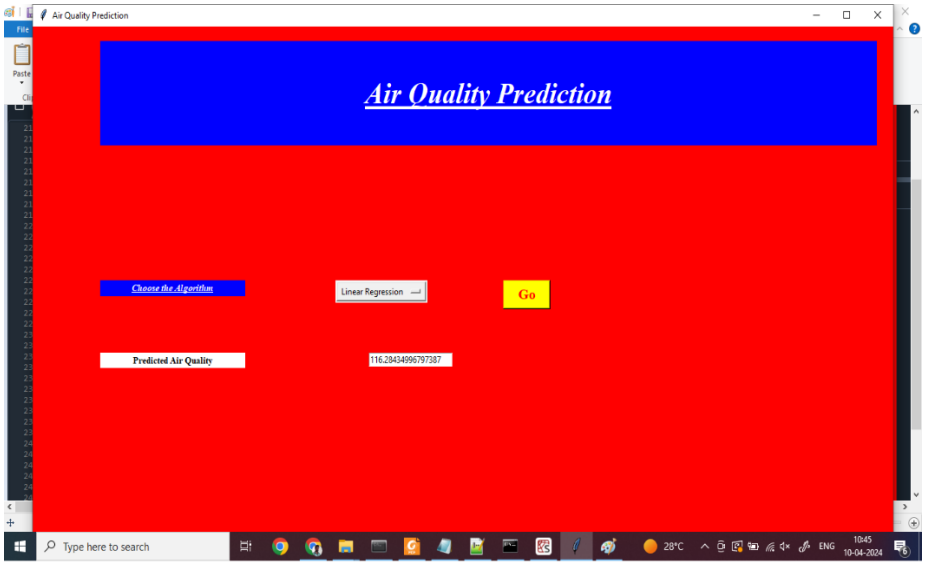
The following snapshots define the results or outputs that we will get after step by step execution of all the modules of the system.

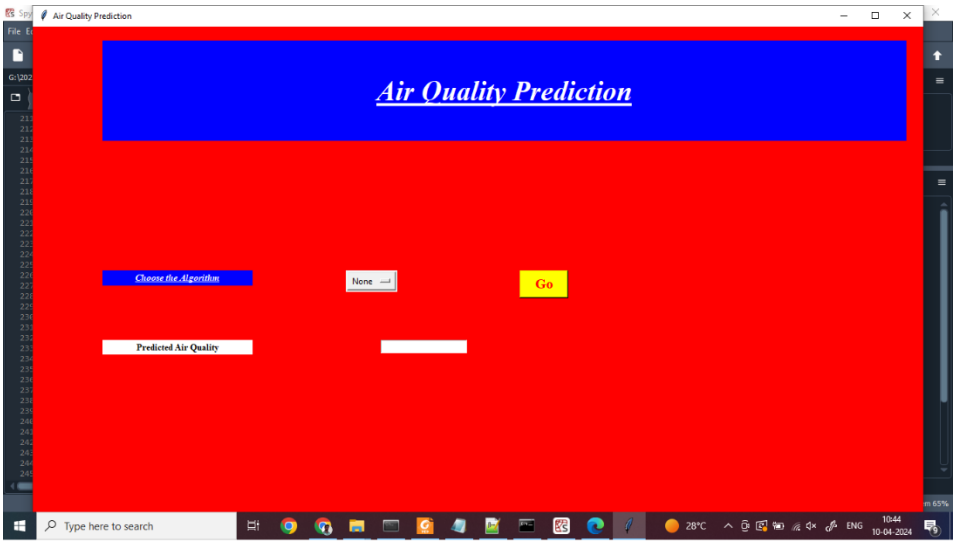
Interpretation:

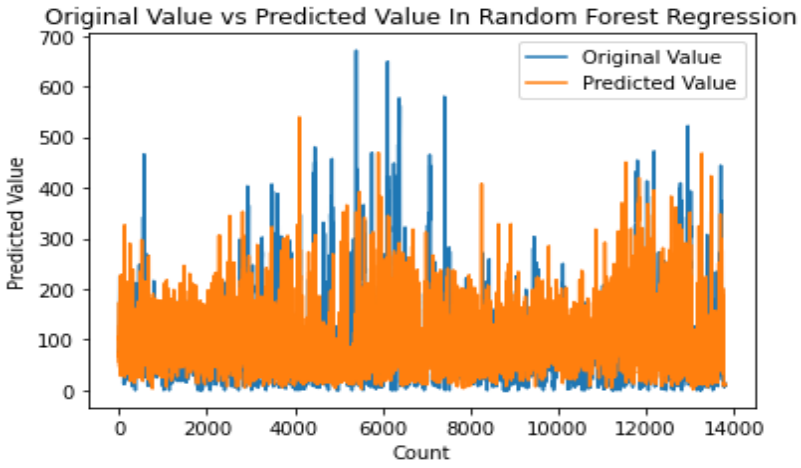


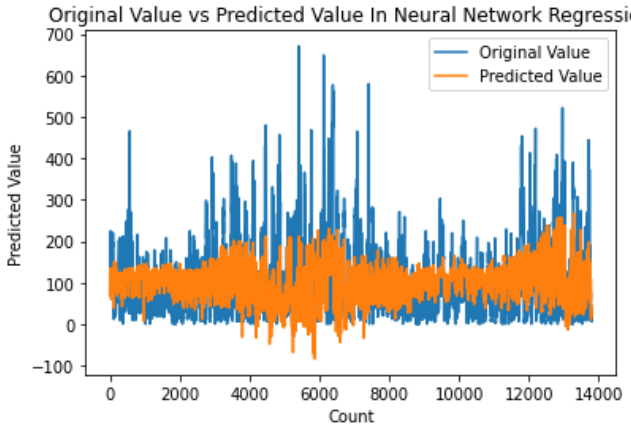


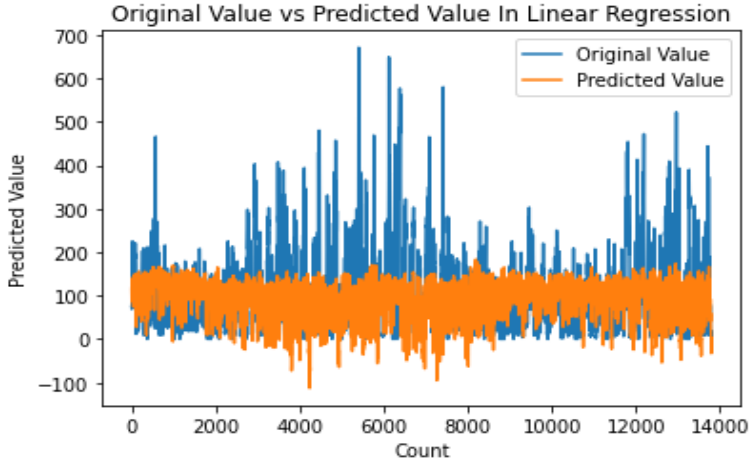


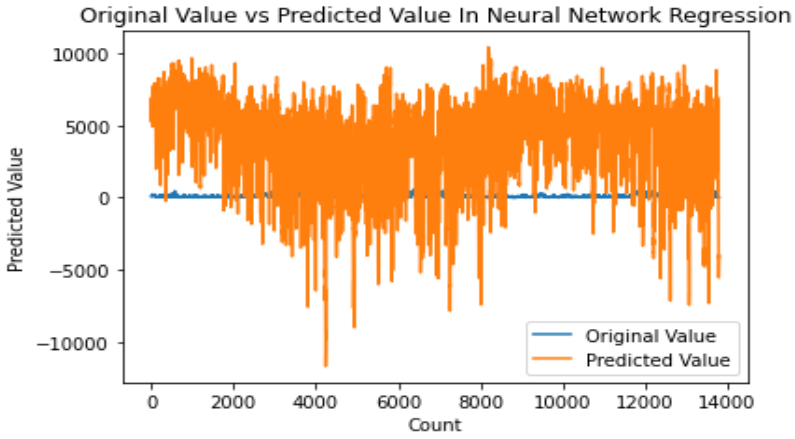

Air Quality Prediction
Air Quality Prediction
Choose the Algorithm
Linear Regression
Go
Predicted Air Quality
116.28434996797387
Type here to search


Air Quality Prediction
Air Quality Prediction
Choose the Algorithm
None
Go
Predicted Air Quality
Type here to search


Original Value vs Predicted Value In Random Forest Regression
Original Value
Predicted Value
Predicted Value
Count
0
100
200
300
400
500
600
700
0
2000
4000
6000
8000
10000
12000
14000







IX.CONCLUSION

Based on the bar plots plotted we come to the conclusion that some cities are highly polluted and need urgent attention. Also for cities like Pune, Mumbai where concentration of so2 is increasing, we can take measures from now to not face problems later. We used ANN model, Logistic Regression and Random Forest model for predicting values of so2. Features such as location_monitoring_station or station code were of no use as they have nothing to do with so2 predictions.

Future Work

This model further, also makes us aware of the challenges in future and research needs such as pm2.5, AQI, etc.

In future, we can take the real time sensors like MQ series sensor, Dust sensor, DHT sensor etc to take the values from real time values and predict the quality of air with the help of IoT Concept.

REFERENCES

[1] Okokpujie, Kennedy & Noma-Osaghae, Etinosa & Odusami, Modupe & John, Samuel & Oluwatosin, Oluga. (2018). A Smart Air Pollution Monitoring System. International Journal of Civil Engineering and Technology. 9. 799-809.
[2] C R, Aditya & Deshmukh, Chandana & K, Nayana & Gandhi, Praveen & astu, Vidyav. (2018). Detection and Prediction of Air Pollution using Machine Learning Models. International Journal of Engineering Trends and Technology. 59. 204-207. 10.14445/22315381/IJETT-V59P238.
[3] Liu, Huixiang & Li, Qing & Yu, Dongbing & Gu, Yu. (2019). Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms. Applied Sciences. 9. 4069.10.3390/app9194069.
[4] Arora, Himanshu & Solanki, Arun. (2020). Prediction of Air Quality Index in Metro Cities using Time Series Forecasting Models. Xi'an Jianzhu Keji Daxue Xuebao/Journal of Xi'an University of Architecture & Technology. 12. 3052-3067. 10.37896/JXAT12.05/1721.
[5] Liu, Xinle & Tan, Wenan & Tang, Shan. (2019). A Bagging-GBDT ensemble learning model for city air pollutant concentration prediction. IOP Conference Series: Earth and Environmental Science. 237. 022027. 10.1088/1755-1315/237/2/022027.
[6] “Air Quality Data in India (2015 - 2020)”, Kaggle. 28-July-2020. [Online].Available:www.kaggle.com/rohan rao/air-quality-data-inindia [Accessed: 23-Mar. 2021].