AIR QUALITY INDEX PREDICTION

USING MACHINE LEARNING

by

Rajarshi SinhaRoy And Swarnendu Sarkhel

A Student Of

B.Sc. in Computer Science

Government General Degree College Singur,
Singur

2020

Abstract

We forecast the air quality by using machine learning to predict the air quality index of a given area. Air quality index is a standard measure used to indicate the pollutant (so2, no2, pm 2.5, pm 10. etc.) levels over a period. We developed a model to predict the air quality index based on historical data of some previous years and predict the Air quality index respect to their pollutant. we improve the efficiency of the model by applying several Estimation Problem logics. Our model will be capable for successfully predicting the air quality index of a total county or any state or any bounded region provided with the historical data of pollutant concentration. In our model by implementing the proposed parameter-reducing formulations, we achieved better performance than the standard regression models. our model has 95% to 98% accuracy on predicting the current available dataset on predicting the air quality index.

Preface

This thesis contains research conducted by the candidate, Rajarshi SinhaRoy and Swarnendu Sarkhel, under the supervision of Prof. Samit Bhanja. The air quality and meteorological data sets used in this study were provided by Prof. Samit Bhanja. The data set was reproduced using the station data from Environment of a specific area. The professors of our college provided the original research topic, direction and critical feedback on the research methods. The development of statistical air quality models and the analysis of results were primarily the work of the candidate, but Prof. Samit Bhanja contributed substantially by suggesting specialized analysis techniques, by helping to interpret the results. Currently no part of this thesis has been published.

Table of Contents

Acknowledgements

I would like to thank Prof. Samit Bhanja for his never-ending guidance and support. The time that was put into this research could not have been done without his coordination and exceptional knowledge of Computer Science and machine learning methods. His constant support and considerate attitude allowed the completion of this research without any pressure or tension. I would also like to thank Prof. Sandipan Basu and Prof. Anupam Sen for his commitment to learning and advisement for this research. I would also like to acknowledge and thank the member of the group Swarnendu Sarkhel. Not only he is my fellow student who helped and supported, but he is also my friend. I would like to thank Prof. Samit Bhanja, for providing data and detailed instruction in each phase of the research, for approving my research and inspiring me through their excellent experience with Environment and I would like to thank for giving me this opportunity, and also the knowledge, gained was able to contribute to my research. Lastly, I would like to thank my parents and all of my friends for their love and encouragement, they are always my best support no matter where I am.

Date : 29/09/2020

Place : Singur

Rajarshi SinhaRoy
And
Swarnendu Sarkhel

Signature

This is to certify that the Project Report entitle "Air Quality Index Prediction Using Machine Learning" submitted by Rajarshi SinhaRoy and Swarnendu Sarkhel to Government General Degree College, Singur is a record of Project work carried out by him under my supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Science in Computer Science of the University of Burdwan.

Date : 29/09/2020 ……

Place : Singur Signature of the Prof.

teacher Samit Bhanja

Assistant Professor,
Department of Computer Science
GGDC, Singur, Hooghly, WB

Introduction

As the largest growing industrial nation, we are producing record amount of pollutants specifically CO2, PM2.5 etc. and other harmful aerial contaminants. Air quality of a particular state or a country is a measure on the effect of pollutants on the respected regions, as per the air quality standard pollutants are indexed in terms of their scale, these air quality indexes indicates the levels of major pollutants on the atmosphere. There are various atmospheric gases which causes pollution on our environment. Each pollution has individual index and scales at different levels. The major pollutants Such as (NO2, SO2, PM 2.5, PM 10) indexes AQI is acquired, with this individual AQI, the data can be categorized based on the limits. We collected the data from the government database, which contains pollutant concentration occurring at various places across. We start by calculating the individual index of the pollutant for every available datapoints and find their respective AQI for the region. We have designed a model to predict the air quality index of every available data points in the dataset, our model is capable of forecasting the air quality of in any given area. By predicting the air quality index, we can backtrack the major pollution causing pollutant and the location affected seriously by the pollutant. With this forecasting model, various knowledge about the data are extracted using various techniques to obtain heavily affected regions on a particular region(cluster). This give more information and knowledge about the cause and seniority of the pollutants.

Problem Statement

The Problem Statement of this project is we forecast the air quality by using machine learning to predict the air quality index of a given area based on its historical data of the pollutants of some previous years and predict the Air quality index respect to their pollutants.

Research Objectives

The research goal of this study is to develop a non-linear updatable model for real-time air quality forecasting, to potentially replace the models currently being used. The ultimate goal is to improve air pollution forecasting in India and in other countries.

Impact of This Project

This project covering the topics related to air quality forecasting, machine learning techniques and updatable model, for an area. Background theory on various machine learning and air quality topics will be covered in this project. The reviewed air quality forecasting studies as well as the modelling techniques will be discussed on how they can be applied to this research for further work. This Project describes the study area and the data sets efficiently thus it shows the real condition of that Environment. The results from all developed forecast models for each pollutant are discussed in detail. The thesis also concludes the original research objectives and recommendations for future research.

Background Knowledge

Air pollution is the introduction of particulates, biological molecules, or other harmful materials into the Earth's atmosphere, causing disease, death to humans, damage to other living organisms such as food crops, or damage to the natural or man-made environment. An air pollutant is a substance in the air that can have adverse effects on humans and the ecosystem. The substance can be solid particles, liquid droplets, or gases. Pollutants are classified as primary or secondary. Primary pollutants are usually produced from a process, such as ash from a volcanic eruption. Other examples include carbon monoxide gas from motor vehicle exhaust, or Sulphur dioxide released from factories. Secondary pollutants are not emitted directly. Rather, they form in the air when primary pollutants react or interact.

Ground level ozone is a prominent example of a secondary pollutant. The six "criteria pollutants" are ground level ozone (O3), fine particulate matter (PM2.5), carbon monoxide (CO), nitrogen dioxide (NO2), sulphur dioxide (SO2), and lead, among which ground level O3, PM2.5 and NO2 (main component of NOx) are the most widespread health threats. Ground level O3, a gaseous secondary air pollutant formed by complex chemical reactions between NOx and volatile organic compounds (VOCs) in the atmosphere, can have significant negative impacts on human health (Chen et al., 2007; Brauer and Brook, 1997). Prolonged exposure to O3 concentrations over a certain level may cause permanent lung damage, aggravated asthma, or other respiratory illnesses. Ground level O3 can also have detrimental effects on plants and ecosystems, including damage to plants, reductions of crop yield, and increase of vegetation vulnerability to disease (EPA, 2005).

Particle pollution (also called particulate matter or PM) is the term for a mixture of solid particles and liquid droplets found in the air. Some particles, such as dust, dirt, soot, or smoke, are large or dark enough to be seen with the naked eye. Others are so small they can only be detected using an electron microscope. Fine particulate matter (PM2.5) consisting of particles with diameter 2.5μm or smaller, is an important pollutant among the criteria pollutants. The microscopic particles in PM2.5 can penetrate deeply into the lungs and cause health problems, including the decrease of lung function, development of chronic bronchitis and nonfatal heart attacks. Fine particles can be carried over long distances by wind and then deposited on ground or water through dry or wet deposition. The wet deposition is often acidic, as fine particles containing sulfuric acid contribute to rain acidity, or acid rain. The

effects of acid rain include changing the nutrient balance in water and soil, damaging sensitive forests and farm crops, and affecting the diversity of ecosystems. PM2.5 pollution is also the main cause of reduced visibility (haze) (EPA, 2005).

Nitrogen dioxide (NO2) is one of a group of highly reactive gases known as "nitrogen oxides" (NOx). US Environmental Protection Agency (EPA) Ambient Air Quality Standard uses NO2 as the indicator for the larger group of nitrogen oxides. NO2 forms quickly from emissions of automobiles, power plants, and on-road equipment. In addition to contributing to the formation of ground-level ozone, and fine particle pollution, current scientific evidence links short-term NO2 exposures, ranging from 30 minutes to 24 hours, with adverse respiratory effects including airway inflammation in healthy people and increased respiratory symptoms in people with asthma (EPA, 2005).

The Air Quality Health Index (AQHI) is a public information tool designed to help understand the impact of air quality on health. Basically, the AQHI is defined as an index or rating scale range from 1 to 10+ based on mortality study to indicate the level of health risk associated with local air quality (Chen and Copes, 2013). The higher the number, the greater the health risk and the need to take precautions. The formulation of Indian national AQHI is based on three-hour average concentrations of ground-level ozone (O3), nitrogen dioxide (NO2), and fine particulate matter (PM 2.5). The AQHI is calculated on a community basis, each community may have one or more monitoring stations and the average concentration of 3 substances is calculated at each station within a community for the 3 preceding hours. AQHI is a meaningful index protecting residents on a daily basis from the negative effects of air pollution. Our study gives direction to predicting individual pollutants of one-hour average concentration instead of AQHI (or its maximum) as the formulation of AQHI is based on health-related science and may evolve over time. Building a forecast system based on individual pollutants and one-hour average concentration will make it more flexible to future changes in health indices. Our result can also be beneficial to external clients and meteorologists.

The concentration of air pollutants including ground level ozone, PM2.5 and NO2 varies depending on meteorological factors, the source of pollutants and the local topography (Dominick et al., 2012). Among these three factors, the one which most strongly influences variations in the ambient concentration of air pollutants is meteorological factors (Banerjee and Srivastava, 2009). Meteorological factors experience complex interactions between various processes such as emissions, transportation and chemical transformation, as well as wet and dry depositions (Seinfeld and Pandis, 1997; Demuzere et al., 2009). In addition, the

spatial and temporal behavior of wind fields are affected by the surface roughness and differences in the thermal conditions (Okeet al., 1989; Roth, 2000), which further influence the dispersion of pollutants. For example, Revlett (1978) and Woland Lioy (1978) found that ambient ozone concentration not only depended on the ratio and reactivity of precursor species, but also on the state of the atmosphere - the amount of sunlight, ambient air temperature, relative humidity, wind speed, and mixed layer (ML) depth, while Tai (2012) found that daily variations in meteorology as described by the multiple linear regression (MLR) including nine predictor variables (temperature, relative humidity, precipitation, cloud cover, 850-hPa geopotential height, sea-level pressure tendency, wind speed and wind direction) could explain up to 50% of the daily PM 2.5 variability in the US. Hence, meteorological factors play an important role in air pollutant concentrations, also making them difficult to model.

Most current air quality forecasting uses straightforward approaches like box models, Gaussian models and linear statistical models. Those models are easy to implement and allow for the rapid calculation of forecasts. However, they usually do not describe the interactions and non-linear relationship that control the transport and behaviour of pollutants in the atmosphere (Luecken et al., 2006). With these challenges, machine learning methods originating from the field of artificial-intelligence have become popular in air quality forecasting and other atmospheric problems (Comrie, 1997; Hadjiiski and Hopke, 2000; Reich et al., 1999; Roadknight et al., 1997; Song and Hopke, 1996). For instance, several neural network (NN) models have already been used for air quality forecast, in particular for forecasting hourly averages (Kolehmainen et al., 2001; Perezet al., 2000) and daily maximum (Perez, 2001). Although NN have advantages over traditional statistical methods in air quality forecasting, NN-based models still need to improve in order to achieve good prediction performance as effectively and efficiently as possible (Wang et al., 2003). A number of difficulties associated with NN hamper their effectiveness in air quality forecasting. These difficulties include computational expense, multiple local minima during optimization, over-fitting to noise in the data etc. Furthermore, there are no general rules to determine the optimal size of network and learning parameters, which will greatly affect the prediction performance.

Another key consideration of forecast models is their updatability when doing Realtime forecasting. For a forecast model, recently observed data should be used to refine the model. This generally follows a procedure that links the discrepancy between model forecasts and the corresponding latest observation to all or some of the parameters in model. Normally there

are two ways for model updating: batch learning and online learning. Whenever new data are received, batch learning uses the past data together with the new data and performs a retraining of the model, whereas online learning only uses the new data to update the model. Batch learning can be computationally expensive in real-time forecasting as the procedure means repeatedly altering a representative set of parameters calibrated over a long historical record. Linear models are generally easy to update online (Wilson and Vallee, 2002), and even with batch learning, linear models are fast and easy to implement. As for non-linear methods, true online learning is difficult for many formulations such as the non-linear kernel method. Furthermore, short time (daily) update via batch learning is too expensive to implement as a non-linear model tends to have more parameters to train and the training process is much slower compared to linear models. Consequently, there is a need to develop non-linear updatable models for real-time forecasting. This study attempts to use the machine learning algorithm, to forecast air pollutant concentrations in an area. This model has an architecture model, but it can be used for online sequential learning. This model can be used in different research areas and it will produce good generalize performance with generally less learning time compared with traditional model.

Preliminary Concept

Introduction

Air pollution is major threat to health and exerts a wide range of impacts on biological and economic systems. The purpose of this literature review is to justify the research objectives of this study in light of previous work by investigating past air quality prediction studies and determining where future research is needed. Literature related to air quality prediction and various types of machine learning methods used in this study are reviewed. Machine learning theory and past applications are examined to show why these methods are likely to perform well in air quality forecasting.

Machine Learning Techniques

Machine learning is a major sub-field in computational intelligence (also called artificial intelligence). Its main objective is to use computational methods to extract information from data. Machine learning has a wide spectrum of applications including handwriting and speech recognition, robotics and computer games, natural language processing, brain-machine interface and so on. In the environmental sciences, machine learning methods have been heavily used in data processing, model emulation, weather and climate prediction, air quality forecasting, oceanographic and hydrological forecasting. (Hsieh, 2009).

Neural Network

Neural network (NN) methods were originally developed from investigations into human brain function and they are adaptive systems that change as they learn (Hsieh and Tang, 1998). There are many types of NN models, the most common one is the multi-layer perceptron (MLP) NN model shown in Fig below,

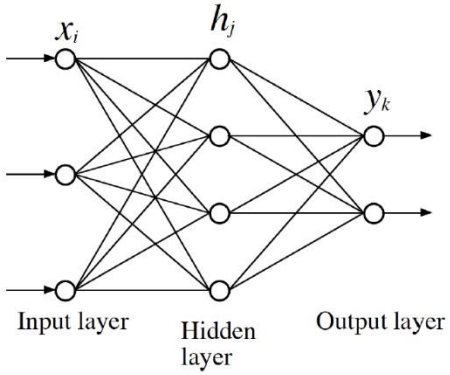


Figure: The general structure of a MLP NN model (Hsieh, 2009).

The input variables xi are mapped to a layer of intermediate variables known as "hidden neuron" hj by $hj = f(\sum_i wjixi + bj)$

and then onto the output variables yk by

$$yk = g(\sum_j \beta kjhj + \beta k0)$$

where f and g are "activation" functions in the hidden layer and the output layer, respectively. Normally f can be the logistic sigmoidal or hyperbolic tangent function and g can be linear in NN models for regression. wji and βkj are weight parameters and bj and βk0 are offset parameters. Their optimal values are learned by model training (Hsieh and Tang, 1998) where the mean squared error of the model output is minimized.

Numerous studies show NN models have good forecasting performance. One of the main challenges in developing a NN model is how to address the problem of over-fitting. An over-fitted NN model could fit the data very well during training, but produce poor forecast results during testing (Hsieh and Tang, 1998). Over-fitting occurs when a model fits to the noise in the data and it will not generalize well to new data sets as shown in Fig below

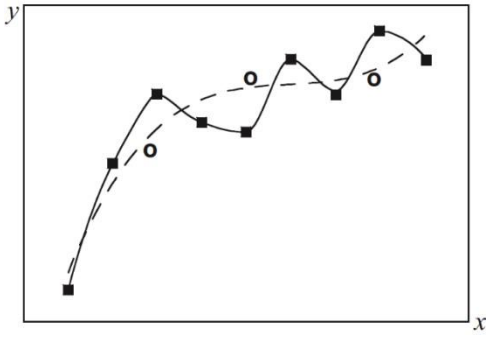


Figure: A diagram illustrating the problem of over-fitting.

The dash curve shows a good fit to noisy data (squares), while the solid curve illustrates over-fitting, where the fit is perfect on the training data (squares), but is poor on the test data (circles) (Hsieh and Tang, 1998)

Typically, regularization (i.e. the use of weight penalty) is used to prevent over-fitting (Golub et al., 1979; Haber and Oldenburg, 2000). This usually requires some of the training data to be used as validation data to determine the optimal regularization parameter to prevent over-fitting. Yuval (2000) introduced generalized cross validation (GCV) to control

overfitting/underfitting automatically in MLP NN and applied the method to forecasting the tropical Pacific SST anomalies. Yuval (2001) used bootstrap resampling of the data to generate an ensemble of MLP NN models and used the ensemble spread to estimate the forecast uncertainty.

Another issue is the computational expense involved during the training process in neural networks. Training the NN model to learn from the target data, we need to minimize the objective function J, defined here to be mean squared error (MSE) between the model output y and the target t. Normally the back-propagation algorithm is used to perform the training tasks, using a gradient-descent approach to reduce the MSE iteratively (Hsieh, 2009), which could be time-consuming.

Air Quality Forecasting Models

An air quality model is a numerical tool used to describe the causal relationship between emissions, meteorology, atmospheric concentration, deposition and other factors. It can give a complete deterministic description of the air quality problem (Nguyen, 2014). The most commonly used air quality models include dispersion models, photochemical models and regression models. Various neural network models, as non-linear regression models, have also been shown to be effective in air quality forecasting. In this section, different models and their applications will be
introduced.

Regression Models

Both linear regression and non-linear regression models have been employed for air quality forecasting. The general purpose of a linear regression model is to learn about the linear relationship between several independent variables (predictors) and a dependent variable (predictand).

Prybutok et al. (2000) built a simple linear regression model for forecasting the daily peak O3 concentration in Houston. The final model used four meteorological and O3 precursor parameters: O3 concentration at 9:00 a.m., maximum daily temperature, average NO2 concentration between 6:00 a.m. and 9:00 a.m. and average surface wind speed between 6:00 a.m. and 9:00 a.m. The correlation coefficient r of this model was 0.47. Chaloulakou et al.

(1999) proposed a multiple regression model to forecast the next day's hourly maximum O3 concentration in Athens, Greece. The set of input variables consisted of eight meteorological parameters and three persistence variables, which were the hourly maximum O3 concentrations of the previous three days. Testing this linear regression model on four separate test data sets, the mean absolute error (MAE) ranged from 19.4% to 33.0% of the corresponding average O3 concentrations.

Non-linear regression models are superior to simple linear regression models because they capture the non-linear relationships between air pollutant and meteorological parameters. Bloom field et al. (1996) described a non-linear regression model to explain the effects of meteorology on O3 in the Chicago area. The model input variables consisted of a seasonal term, a linear annual trend term, and twelve meteorological variables. The observed ozone and meteorological data in 1981-1991 were divided into subsets for model development and validation. The model error was within ± 5 ppb about half the time, and within ± 16 ppb about 95% of the time. Bloom field et al. (1996) demonstrated that the meteorological data accounted for at least 50% of the ozone concentration variance.

Neural Network Models

Although many approaches such as box models, Gaussian plume models, persistence and regression models are commonly applied to characterize and forecast air pollutants concentration, they are relatively straightforward with significant simplifications (Luecken et al., 2006).

A promising alternative to these models is the neural network model (Lal and Tripathy, 2012; Nejadkoorki and Baroutian, 2012; Gardner and Dorling, 1998). Several NN models have already been used for different air pollutant concentration forecast. Gardner and Dorling (2000) used MLP NN to forecast the hourly ozone concentration at five cities in UK and they found that NN outperformed both CART (classification and regression tree) and linear regression (LR). The predictors used included the amount of low cloud, base of lowest cloud, visibility, dry bulb temperature, vapour pressure, wind speed and direction. To account for seasonal effect, they had two extra predictors in model 2, sin(2_d=365) and cos(2_d=365), with d the Julian day of the year, thereby informing the model where in the annual cycle the forecast was made. Ballester et al. (2002) used a finite impulse response NN model to make

1-day advance predictions of 8-hr average ozone concentrations in eastern Spain. The input variables were observed 2h lagged observed values of air quality and meteorological inputs. The models were evaluated using data from the 1996 to 1999 ozone seasons (July to September). The statistics of the model fits for three sampling sites ranged from 6.39 to 8.8 ppb for MAE and from 0.73 to 0.79 for R.

Several authors compared different approaches when applied to different pollutants and prediction time lags (Boznar et al., 1993; Lu and Wang, 2005; Yi and Prybutok, 2002). In the overview of NN application in the atmospheric sciences, Gardner and Dorling (1998) concluded that NN generally gives as good or better results than linear methods

Study on Data

Study Area

The updatable model output statistics - air quality (UMOS-AQ) model uses observations from more than 250 station. The stations belong to the National Air Pollution Surveillance Network, where each station measures all or a combination of the concentrations of ozone (O3), fine particulates and nitrogen dioxide (NO2), Nitrogen Monoxide, Sulphur Dioxide, PM2.5, PM10 etc. Several stations are used to collect the data from big cities and some stations are used in coastal cities and the major center for oil and gas industry and some background cities. They all have different topography, weather conditions and major pollution sources.

Dataset

The observational air pollutant data were from automated near-real-time (NRT) hourly reports of local Pollutant-concentrations from urban and rural AQ measurement stations.

Index	Month	GMT	RNO	RNO2	RNXO	RO3	RPM10	RPM25	RSO2
0	1.1.10	0:00	63.8742	54.7677	118.758	15.329	27.3774	23.4645	4
1	1.1.10	1:00	51.6903	48.6161	100.4	17.2968	25.6581	22.5774	3.6129
2	1.1.10	2:00	44.0258	43.271	87.4097	19.7258	24.471	20.9065	3.27419
3	1.1.10	3:00	42.1419	41.3032	83.5613	20.1613	23.9419	20.7548	3.33226
4	1.1.10	4:00	47.5355	42.4645	90.1065	19.1	23.229	20.371	3.44194
5	1.1.10	5:00	68.2032	49.5032	117.868	15.7742	23.9484	21.1516	3.79677
6	1.1.10	6:00	108.116	62.9774	171.316	11.7548	25.4839	22.8065	4.4871
7	1.1.10	7:00	140.232	73.4774	214.006	10.0935	26.9129	23.2452	5.22258
8	1.1.10	8:00	146.632	75.6065	222.555	10.229	28.5774	23.3452	5.60968
9	1.1.10	9:00	142.474	73.5	216.258	12.1968	30.5516	24.3806	5.58387
10	1.1.10	10:00	133.429	70.1903	203.91	14.9323	31.3645	24.571	5.53226

Fig: Dataset (Hourly) including all pollutants, months and time of taking data.

There are 7 pollutants and data sample no are 2592. The first 100 pollutants-raw data plot fig is in below :

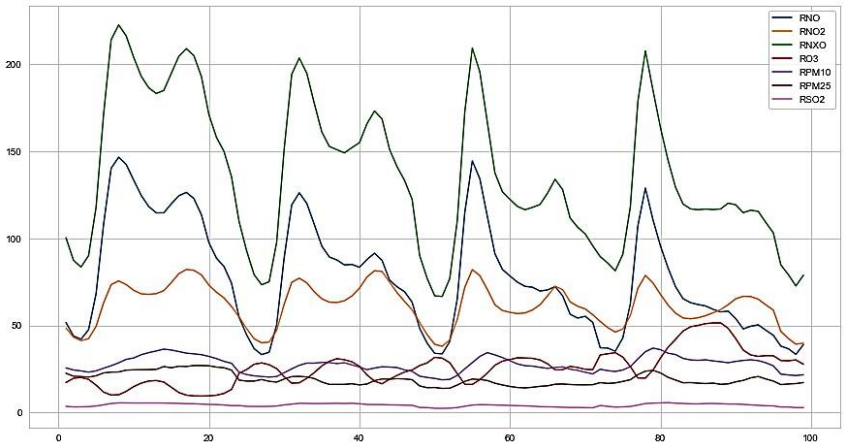


Fig: Plot of first 100 data

.

Methodology

Input Data Pre-processing

In this dataset the outliers are mainly of faulty sensor or transmission errors, these errors have huge variation than the normal valid results. We know the standard range of pollutants occurs on a particular area so to remove the outliers from the data we use boundary value analysis (BVA). By using BVA we found the upper quartile range and lower quartile range of a given data.

AQI Calculation

We acquired the dataset with various columns of sensor data from various places. we have the average readings of ambient air quality with respect to air quality parameters, like Sulphur dioxide (SO2), Nitrogen dioxide (NO2), Particulate Matter 2.5 and Particulate Matter 10 etc. Data acquired from the source has more noisy data since few of the data from the stations have been shifted or closed the period were marked as

NAN or not available.so we have to pre-process the data in order to remove the outliers. Each individual pollutant indexes, gives the relationship between the pollutant concentration and their corresponding individual index.

```
def calculate_ss(RSO2):
    ss=0
    if (RSO2<=40):
     ss= RSO2*(50/40)
    if (RSO2>40 and RSO2<=80):
     ss= 50+(RSO2-40)*(50/40)
    if (RSO2>80 and RSO2<=380):
     ss= 100+(RSO2-80)*(100/300)
    if (RSO2>380 and RSO2<=800):
     ss= 200+(RSO2-380)*(100/800)
    if (RSO2>800 and RSO2<=1600):
     ss= 300+(RSO2-800)*(100/800)
    if (RSO2>1600):
     ss= 400+(RSO2-1600)*(100/800)
    return ss


#here it's calculated on redundent data.
data['ss']=data['RSO2'].apply(calculate_ss)
newdata['ss']=newdata['RSO2'].apply(calculate_ss)
df= data[['RSO2','ss']]
ndf=newdata[['RSO2','ss']]
```

Fig: shows an example of the individual pollutant index calculation of SO2

Now AQI calculation based on the individual pollutant index and it is calculated as per Indian Govt. Standards

```
#function to calculate the air quality index (AQI) of every data value
#its is calculated as per indian govt standards

def calculate_aqi(ss,ni,spi,rpi,na,nx,oi):
    aqi=0
    if(ss>ni and ss>spi and ss>rpi and ss>na and ss>nx and ss>oi):
     aqi=ss
    if(spi>ss and spi>ni and spi>rpi and spi>na and spi>nx and spi>oi):
     aqi=spi
    if(ni>ss and ni>spi and ni>rpi and ni>na and ni>nx and ni>oi):
     aqi=ni
    if(rpi>ss and rpi>ni and rpi>spi and rpi>na and rpi>nx and rpi>oi):
     aqi=rpi
    if(na>ss and na>ni and na>spi and na>rpi and na>nx and na>oi):
     aqi=na
    if(nx>ss and nx>ni and nx>spi and nx>rpi and nx>na and nx>oi):
     aqi=nx
    if(oi>ss and oi>ni and oi>spi and oi>na and oi>nx and oi>rpi):
     aqi=oi

    return aqi

data['AQI']=data.apply(lambda x:calculate_aqi(x['ss'],x['ni'],x['spi'],x['rpi'], x['na'], x['nx'], x['oi']),axis=1)
df= data[['Month','GMT','ss','ni','rpi','spi','na','nx','oi','AQI']]
df.head()
```

Fig: AQI calculation

Index	Month	GMT	ss	ni	rpi	spi	na	nx	oi	AQI
0	1.1.10	0:00	5	100.96	39.1075	27.3774	112.343	138.758	19.1613	138.758
1	1.1.10	1:00	4.51613	93.2702	37.629	25.6581	97.1129	120.4	21.621	120.4
2	1.1.10	2:00	4.09274	86.5887	34.8441	24.471	87.5323	107.41	24.6573	107.41
3	1.1.10	3:00	4.16532	84.129	34.5914	23.9419	85.1774	103.561	25.2016	103.561
4	1.1.10	4:00	4.30242	85.5806	33.9516	23.229	91.9194	110.106	23.875	110.106
5	1.1.10	5:00	4.74597	94.379	35.2527	23.9484	117.754	137.868	19.7177	137.868

Fig: AQI along with pollutants and individual pollutant index

Now the data plot of the AQI is in the figure below

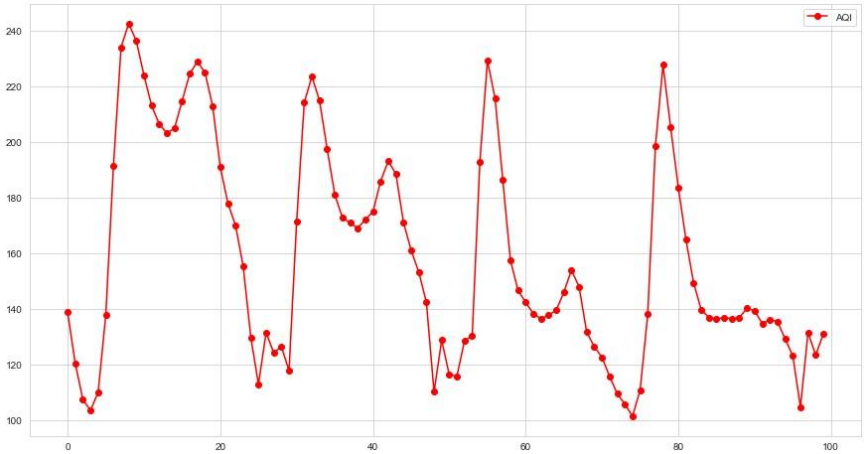


Fig: Graph between AQI and sample data

Pearson Correlation Coefficient (r)

The Pearson correlation coefficient, reflecting the degree of linear relationship between two variables, is defined by

$$r = \frac{\operatorname{cov}(\hat{Y}, Y)}{\sigma_{\hat{Y}}\,\sigma_{Y}}$$

where $\hat{Y}$ demotes the model predicted pollutant concentrations, Y the observed values, cov the covariance and σ the standard deviation. This coefficient varies from 1 to 1, with 0 indicating no relationship. While the Pearson correlation is a good measure of the linear association between predictions and observations, it does not take into account the prediction bias, and is sensitive to rare extreme events.

```
PSS=data[['RNO','RNO2','RNXO','RO3','RPM10','RPM25','RSO2']]
sb.pairplot(PSS)
corr = PSS.corr()
corr
```

Index	RNO	RNO2	RNXO	RO3	RPM10	RPM25	RSO2
RNO	1	0.808914	0.955208	-0.486873	0.565506	0.402057	0.52353
RNO2	0.808914	1	0.884359	-0.301948	0.690474	0.515651	0.493918
RNXO	0.955208	0.884359	1	-0.492486	0.600101	0.422191	0.554069
RO3	-0.486873	-0.301948	-0.492486	1	-0.178849	-0.307877	-0.254437
RPM10	0.565506	0.690474	0.600101	-0.178849	1	0.868501	0.365362
RPM25	0.402057	0.515651	0.422191	-0.307877	0.868501	1	0.318274
RSO2	0.52353	0.493918	0.554069	-0.254437	0.365362	0.318274	1

Fig: Pearson Correlation Coefficient code and result

From Pearson Correlation Coefficient, we can identify the efficiently co-related pollutants. NO, NO2 are co-related with NXO and PM10 is co-related with PM2.5. Now we delete those pollutant data from data set and create a new dataset. After that we again calculate the individual pollutant index and using that we find the AQI.

Index	Month	GMT	RNXO	RO3	RPM25	RSO2	ss	rpi	nx	oi	AQI
0	1.1.10	0:00	118.758	15.329	23.4645	4	5	39.1075	138.758	19.1613	138.758
1	1.1.10	1:00	100.4	17.2968	22.5774	3.6129	4.51613	37.629	120.4	21.621	120.4
2	1.1.10	2:00	87.4097	19.7258	20.9065	3.27419	4.09274	34.8441	107.41	24.6573	107.41
3	1.1.10	3:00	83.5613	20.1613	20.7548	3.33226	4.16532	34.5914	103.561	25.2016	103.561
4	1.1.10	4:00	90.1065	19.1	20.371	3.44194	4.30242	33.9516	110.106	23.875	110.106
5	1.1.10	5:00	117.868	15.7742	21.1516	3.79677	4.74597	35.2527	137.868	19.7177	137.868

Fig: new AQI after Pearson Correlation Coefficient

```
gp=data['AQI']
ngp=newdata['AQI']
Mn=data['Month']
plt.plot(Mn,gp,label ="RAW DATA AQI")
plt.plot(Mn,ngp,label ="REDUNDENT DATA AQI")

plt.xlabel('Month')
plt.ylabel('AQI')
plt.legend(loc='upper right')
plt.title('AQI vs Redundent AQI')
plt.show()
```

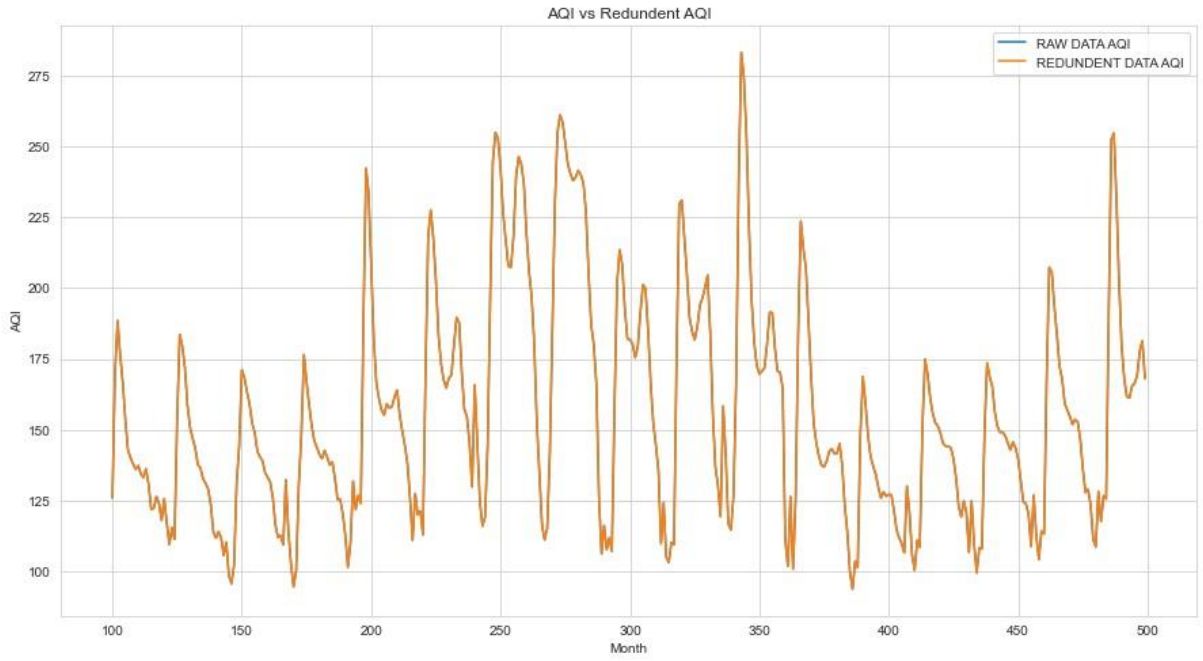


Fig: AQI vs Redundant AQI graph plot

Dynamic Time Wrapping

In time series analysis, dynamic time warping (DTW) is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed. For instance, similarities in walking could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and graphics data — indeed, any data that can be turned into a linear sequence can be analysed with DTW. A well-known application has been automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. It can also be used in partial shape matching application.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restriction and rules:

- *Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa*
- *The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)*
- *The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)*
- *The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa, i.e. if j > i are indices from the first sequence, then there must not be two indices l > k in the other sequence, such that index i is matched with index l and index j is matched with index k , and vice versa.*

```
#dynamic time wraping.....


x = newdata['AQI']
y = data['AQI']
distance, path = fastdtw(x, y, dist=euclidean)
print(distance)
plt.plot(path)
```

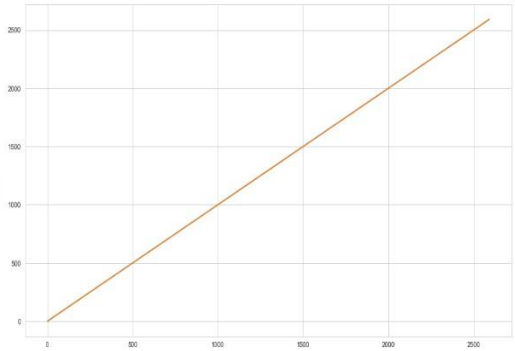


Fig: Dynamic Time Wrapping code and graph

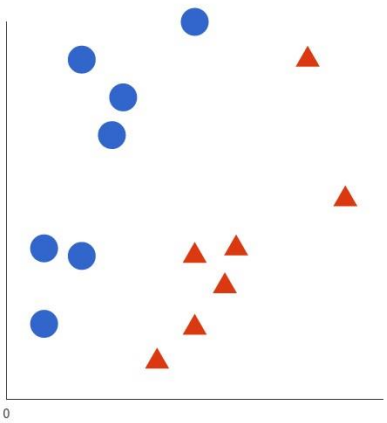
Split the Dataset

Using this Forecast approach, we spitted the dataset into two parts of first 70% and rest 30% data into test and train datasets to identify the huge seasonal variations and trend.

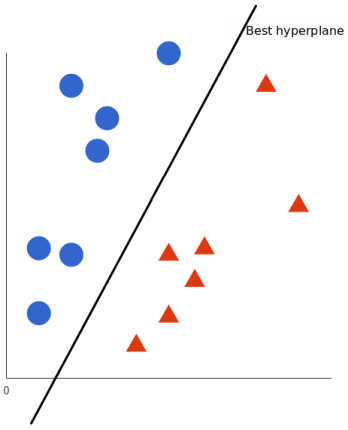
Support vector Machine (SVM)

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labelled training data for each category, they're able to categorize new text.

The basics of Support Vector Machines and how it works are best understood with a simple example. Let's imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x, y) coordinates, outputs if it's either red or blue. We plot our already labelled training

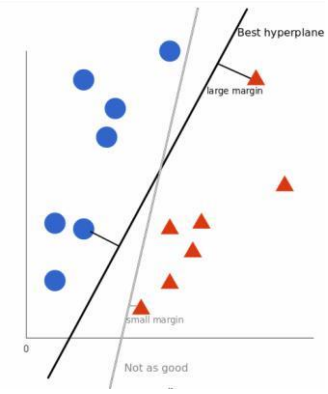


data on a plane:



A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

But what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.



```
from sklearn.model_selection import train_test_split
X_train , X_test,y_train, y_test = train_test_split(X,y,test_size = 0.3)

from sklearn.svm import SVC
sv= SVC(kernel = 'linear')
sv.fit(X_train,y_train)


y_pred = sv.predict(X_test)
```

Fig: Code for SVM, Splitting the dataset and predict the AQI

Mean Absolute Error (MAE)

The mean absolute error (MAE) is the average absolute value of the forecast errors, with

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |Yi - Yi|$$

where N is the number of data points, Yi is the observed value and Yi *is the predicted value.*

Root Mean Square Error (RMSE)

The root mean squared error (RMSE) is the square root of the mean squared error between the predictions and observations,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2}$$

RMSE is more sensitive to outliers than the MAE.

Accuracy calculation Accuracy

Accuracy calculation Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is essentially an important efficiency parameter and it is without problems. It is the ratio of properly expected commentary to the total number of records. We may think that, if we've got excessive correctness then the mannequin is exceptional.

Results and Discussion

In this Project we are concerned about the accuracy of the model. But we tried a lots of neural networking methods to find the best accuracy to make it more reliable for realtime project. In below we talk about the accuracy , graphs, esstimate Aqi value regarding to the value of pollutants.

Project's graphs are the main key to find the best results .We createv the graph of the given dataset and create a AQI vs monthly graph and then we reduce the data set using pearson co relation co efficient method. After achiving the reduced data set we get the AQI Graph and test both . We get that both are nearly eqaul to eaach other .So both are overlapping the graph.

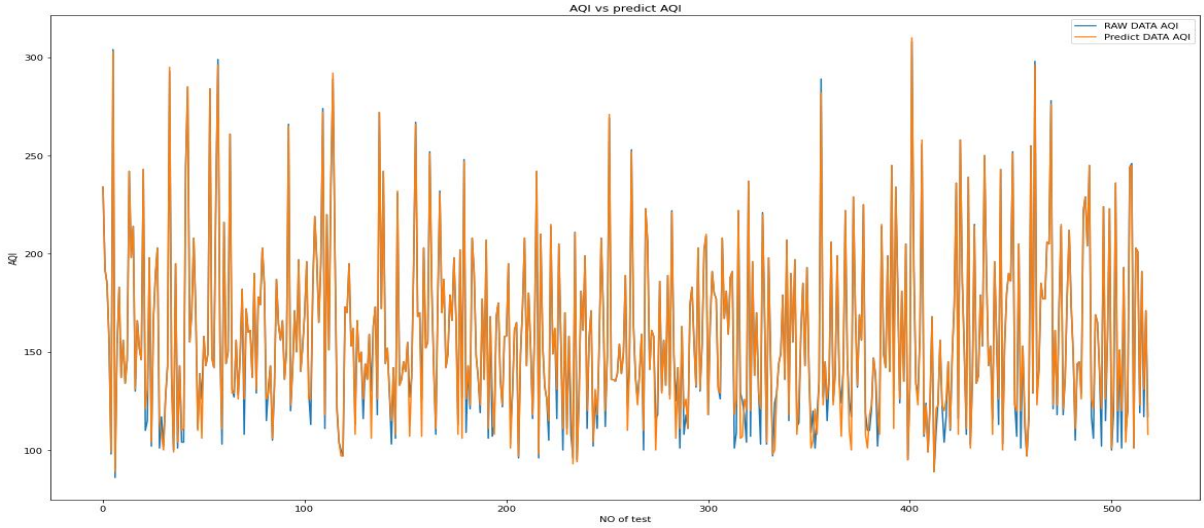
Month vs AQI graph

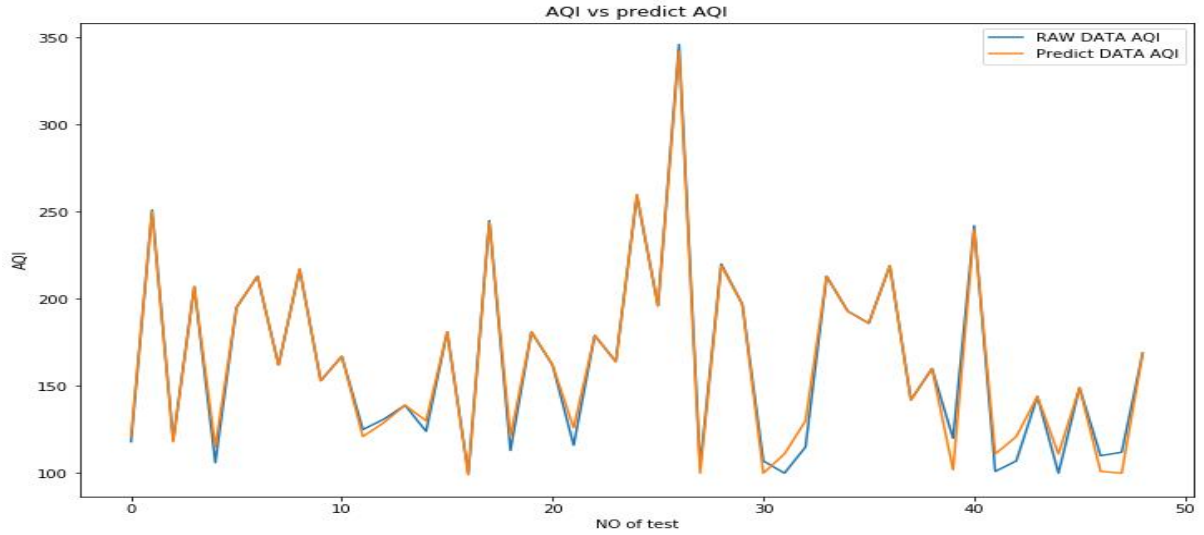
The above is a graph shows AQI with respect to various month after reducing the dataset. In x-axis "Month" is considered and in y-axis "AQI" is considered. This is the first thing we get by calculating and plotting graph.

From here we get that the reducing data set and the original data set is quite equal. So we can perform the rest of the preoject on the reducing data set.

In SVM we tried several karnel to find the best accuracy and graph plot . Only in Liner karnel we get the best accuracy of 95-98% (depends on the project datasize and splitting size). Box plot is one of common graphical systems utilized in EDA.A crate plot or boxplot is a helpful method for graphically portraying gatherings of numerical information through their

quartiles. Box plots may likewise have lines broadening vertically from the containers(bristles) demonstrating inconstancy outside the upper and lower quartiles, henceforth the terms box-and-hair plot and box-and stubblegraph. Exceptions might be plotted as individual focuses. We created a box plotting graph to maintain the results as well as we create a graph . In below graph x-axis is "Number of test" and y-axis is "AQI'. Orange colour is for "Raw Data AQI" and blue colour is for "Predict Data AQI". The graph tells that the line are nearly overlapping in nature.





The above graph is a close view of the previous graph (here take only 50 data). Here we can see nearly overlapping nature of both the dataset. This implies that our prediction is nearly close to the actual value, and our machine is 95%-98% capable of predicting the "AQI" correctly on the basis of range defined from 1-6. However, can be improved by feeding more data.

In India there are six AQI categories, namely Good, Satisfactory, Moderately polluted, Poor, Very Poor, and Severe. The proposed AQI will consider eight pollutants (PM10, PM2.5, NO2, SO2, CO, O3, NH3, and Pb) for which short-term (up to 24-hourly averaging period) National Ambient Air Quality Standards are prescribed.[23] Based on the measured ambient concentrations, corresponding standards and likely health impact, a sub-index is calculated for each of these pollutants. The worst sub-index reflects overall AQI. Likely health impacts for different AQI categories and pollutants have also been suggested, with primary inputs from the medical experts in the group. The AQI values and corresponding ambient concentrations (health breakpoints) as well as associated likely health impacts for the identified eight pollutants are as follows:

AQI Category, Pollutants and Health Breakpoints

AQI Category (Range)	PM_{10} (24hr)	$PM_{2.5}$ (24hr)	NO_2 (24hr)	O_3 (8hr)	CO (8hr)	SO_2 (24hr)	NH_3 (24hr)	Pb (24hr)
Good (0–50)	0–50	0–30	0–40	0–50	0–1.0	0–40	0–200	0–0.5
Satisfactory (51–100)	51–100	31–60	41–80	51–100	1.1–2.0	41–80	201–400	0.5–1.0
Moderately polluted (101–200)	101–250	61–90	81–180	101–168	2.1–10	81–380	401–800	1.1–2.0
Poor (201–300)	251–350	91–120	181–280	169–208	10–17	381–800	801–1200	2.1–3.0
Very poor (301–400)	351–430	121–250	281–400	209–748	17–34	801–1600	1200–1800	3.1–3.5
Severe (401–500)	430+	250+	400-	748+	34+	1600+	1800+	3.5+

By this data analysis we came to know that there are seasonal variations and trend, in order to reduce these metrics, we resample the data month wise to predict it month wise. By resampling the data, we can reduce the outlier more efficiently than raw data. After removing the outlier's SVM is applied to the filtered data and to fit the Trent line on the data points gradient descent hyper parameters are used to optimize the model.

Conclusion

Our main aim was to predict the Air Quality Index (AQI) from its previous values. Predicting AQI will help the society and the people to take action accordingly, such as one can decide whether to go outside with a mask or without a mask or to stay inside. Calculating AQI with too many sensors are costly as well as time consuming here in this project, we reduced the cost without any hamper to its efficiency. We had provided the data in hourly basis so one get information more precisely and act accordingly.

Although our machine can predict up to 96%-98% on range basis, still the machine can further be improved with more data collected from sensors hourly. Another improvement can be done in time management field, here the data is predicted hourly we can further improve to make it predictable in half an hour or maybe in minutes.

Since our model is capable of predicting the current data with 95% accuracy it will successfully predict the upcoming air quality index of any particular data within a given region. With this model we can forecast the AQI and alert the respected region of the country also it a progressive learning model it is capable of tracing back to the particular location needed attention provided the time series data of every possible region needed attention. The air quality information utilized in this paper originates from the china air quality checking and investigation stage, and incorporates the normal every day fine particulate issue (PM2.5), inhalable particulate issue (PM10), ozone (O3), CO, SO2, NO2 fixation and air quality record(AQI).The essential perspectives that should be viewed as with regards to gauging of the poison focus are its different sources alongside the components that impact its fixation.

Future Research

- *India meteorological department wants to automate the detecting the air quality is good or not from eligibility process (real time).*
- *To automate this process by show the prediction result in web application or desktop application.*
- *To optimize the work to implement in Artificial Intelligence environment.*

Related Work

The autoregressive integrated moving average model (ARIMA) is one of the most important and widely used models to forecast time series. Through the years, since the concern with air quality and quality of life in urban areas has emerged, statistical methods like ARIMA have been widely used to forecast the levels of air pollutants and air quality. For instance, the ability of ARIMA to forecast the monthly values for the air pollution index was studied in ,demonstrating that it could produce forecasts that fall under the 95% confidence level. More recently, the performance of ARIMA was compared against a Holt exponential smoothing model to predict AQI daily values .

With the increasing amount of historical data available for analysis and the need for performing more accurate forecasts in different scientific areas and domains, machine learning models have drawn attention, establishing themselves as a solution that can replace the more classical statistical models in time-series forecasting. Specifically, ML algorithms have been widely used to forecast air quality.

Due to the high nonlinear processes that involve the concentrations of pollutants and their partially known dynamics, it is very difficult to produce a model able to forecast these types of events. ML models are an example of nonparametric and nonlinear models that leverage only in historical information to learn the hidden relationship between data. In general, ML approaches, like artificial neural networks (ANNs), genetic programming (GP), and support vector machines (SVMs), have been shown to outperform ARIMA when predicting time series (TS) with a high level of nonlinearity. For instance, Sharda and Patil compared the results achieved by an ANN against ARIMA. Later, Alon et al. compared ANNs against traditional methods, like ARIMA,

Winter exponential smoothing, or multivariate regression, concluding that ANNs outperform the traditional statistical methods when the dataset presents more volatile conditions.

Numerous published contributions exist exploiting the use of support vector machines (SVMs) to forecast time series, and several authors applied SVM to generate models to forecast the air quality and level of pollutants.

Regarding time-series air quality forecasting, Lu and Wang applied SVMs to forecast the air quality in downtown Hong Kong. The results showed that the SVM model delivers more promising results than other ML approaches. Li et al. proposed a hybrid approach model based on co-integration theory, SVM, and the flower pollination algorithm. The results comparing this hybrid model with the particular models show that the hybrid model outperforms and combines all the advantages from each model.

References

- To complete this our project guide Mr. Samit Bhanja has helped us a lot for his guidance this project led to success.
- We used free platform such as youtube, python.org, scikitlearn.org to gain knowledge of how to implement the code work and get desired outcome.
- From website www.airveda.com we learned how to range the data and categorise.
- We downloaded the csv file from website https://datahub.io/london/air-quality and worked on it.
- Calculation is done on the basis of the Indian Method of calculation, we learned that from website https://cpcb.nic.in/air-pollution V. M. Niharika and P. S. Rao, "A survey on air quality forecasting techniques,"International Journal of Computer Science and information Technologies, vol. 5, no. 1, pp.103-107, 2014.
- NAAQS Table. (2015). [Online]. Available: https://www.epa.gov/criteria-air-pollutants/naaqs-table
- E. Kalapanidas and N. Avouris, "Applying machine learning techniques in air quality prediction," in Proc. ACAI, vol. 99, September 2017
- Questioning smart urbanism: Is data-driven governance a panacea? (November 2,2015).[Online].Available: http://chicagopolicyreview.org/2015/11/02/questioningsmart-urbanism-is-data-driven-governance-a-panacea

- Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data,"in Proc. the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2267-2276, August 10, 2015.