AQI Analysis

Dhvani Vashist, Tanishi Tyagi, Tushar Mittal, Vishvesh Gupta
E18CSE048, E18CSE186, E18CSE191, E18CSE213

Department of Computer Science Engineering,
Bennett University.
Greater Noida-201310, Uttar Pradesh, India

***Abstract*—In this report we work on AQI Data available from different cities in India. With the increasing population, air quality is degrading and has become a serious threat to human lives. To avoid these repercussions, we use Air purifiers in our homes. We either turn them on or off according to our requirements which can be a little difficult and uneasy to use or we can simply leave it on. But it will consume a lot of power and it's not the optimal solution. To solve this problem we use machine learning via python and automate the process of turning them on or off by predicting the current AQI of the surroundings. Based on real time data we are able to classify AQI into 5 categories.**

Keywords— AQI, Pollution, Machine Learning, Air Quality, Random Forest,

I. INTRODUCTION

A person breath around 15,000 L of air every day. Breathing clean air is beneficial for our health. The World Health Organization states that breathing cleaner air can lessen the danger of coronary illness, lung cancer, or some other respiratory diseases, for example, asthma, etc. It also increases the quality of life. Over the last few years, air quality has degraded drastically. It is because of various factors like increasing population, traffic, wildfires, burning of fossil fuels, emissions from industries, etc. Because of all these factors, clean air has become a myth these days. The pollutants present in the air like Carbon Monoxide (CO), Nitrogen Dioxide (NO2), volatile organic compounds, fine particulate matter (PM2.5) causes some serious harm to the air quality thus, causing distress to public health. Even the air that we breathe inside our rooms is infused with these invisible particles, which could be detrimental to our health or even aggravate already existing health issues. Studies show that the air we breathe indoors is two to five times more than the outdoors. This is because confined space allows the pollutants to build more than open spaces. We are always exposed to these pollutants, which results in health problems in the long run. Studies show that breathing polluted air is responsible for around 5 million premature deaths every year from diseases like heart attacks, respiratory illness, strokes, etc. Some research shows that continuous exposure to these pollutants can have adverse effects than a pack of cigarettes per day. These statistics are very concerning, and everyone is doing their best to avoid all these repercussions by taking proper precautions.

With the advancement in technology, humans have been able to develop advanced air purifiers which now in this state of a pandemic are just as important as healthy eating or clean water. The main objective of these purifiers is to remove the pollutants present in the surroundings and increase the overall quality of air. To prevent the harmful effects of polluted air we use air purifiers in our home. In order to use these purifiers, we either manually turn it on or off randomly or always keep it on. But these methods consume a lot of energy and thus, are not the optimal solutions. Thus, the main problem with these purifiers is energy consumption. We want an air purifier to be energy efficient while effectively increasing the air quality of its surroundings. To solve this problem, we came up with the idea of predicting optimal time slots for which the air purifier can be turned on so that we can save energy and resources.

In this project, we trained a machine learning model on the past AQI values of different regions. The dataset which we used had concentrations of different pollutants in the surrounding that affect the AQI value of that region. With this model, we predicted the current AQI of a particular region based on real-time values which we extracted from an active website. We further predict the AQI bucket of the region and decide whether we should turn on the purifier or not.

Before starting with the project, we did some research and found publications on similar topics. But there were only a few which worked on a dataset based on India. So we decided to work on a dataset having values of different

regions in India. We also found some projects which were based on indoor air quality, analysis of gasses present in the atmosphere, and some patterns were observed in the analysis, which further helped in predicting air quality in the near future. The most widely used method which was adopted in previous works is the regression model. We have however worked on a classification model. We compared various models and selected the one which gave us the maximum accuracy.

Apart from all this, the distinct feature in our project is the use of real-time values. With the help of web scraping, we extracted real-time values of the concentration of different pollutants in a region from a regularly updated website. And based on those values, we predicted the AQI values and AQI bucket. Also, there's not much work done on AQI and energy consumption by air purifiers before.

II. PREVIOUS RELATED WORK:

Clean air is not only important but also essential to all of us. We all are well versed with the effect that air pollution has on the air quality in our time because of which clean air is close to a myth these days. Articles about deteriorating air quality and issues related to it suggest that improving air quality is the need of the hour. The WHO guideline [6] also hands out the deep information on the four most popular pollutants and offers a thorough overview of the problem and risk and project formulations that are being carried out around the world and [2] are carefully analyzing recorded measurements of gasses present in the atmosphere and some patterns were observed in the analysis which helped in predicting the air quality in near future by establishing certain machine learning methods. For the prediction of air quality using machine learning, several attempts have been made in the past. The most widely used method which was adopted is the regression model. In his paper, Ryan Allen [3] works upon the prediction of indoor particulate matter via measuring them and applying a regression model. Though he wasn't satisfied with the result as the contribution to residential indoor quantities of outdoor particulate matter (PM) was not well known at that time. Since then the availability and accuracy of the data have improved. Owing to the advent of the Internet of Things (IoT), sensor data is attracting growing global interest. To compute forecasts, logic is extended to such sensor data. In their study [4] authors proposed that by resolving sensor data uncertainties and using its discovered data pattern, the combined application of BRBES (Belief rule-based expert systems) and Deep Learning will compute prediction with enhanced accuracy. This paper, therefore, suggests a novel predictive model that is based on the BRBES and Deep Learning combined methodology. Jaehyun Ahn [5] also suggested a similar methodology by developing a microchip made of sensors that are capable of recording measurements regularly and suggested a model that uses deep learning to estimate atmospheric changes. Which may detect trends in the measurements by recording quality measurements and forecast the air quality shortly by analyzing them. In contrast to the above-mentioned papers, if surface sensors are not available, [9] remote satellite sensing methods can also be used. Colin Bellinger's work [10] shows that the field of air pollution epidemiology is increasingly making use of data mining on satellite sensors to assess PM2.5 where surface sensors were not available. Deep dive into the suggested regression models, papers [6] suggest that support vector regression (SVR) and random forest regression (RFR) were among the most recognized models for AQI prediction. To construct regression models for the prediction of the Air Quality Index (AQI) in Beijing and the concentration of nitrogen oxides (NOX) in an Italian city, Huixiang Liu and fellow authors used support vector regression (SVR) and random forest regression (RFR) based on two publicly available datasets. The experimental findings in their paper showed that in the AQI prediction, the SVR-based model (RMSE = 7.666, R2 = 0.9776, and r = 0.9887) performed better and the RFR-based model performed better in the NOX concentration prediction (RMSE = 83.6716, R2 = 0.8401, and r = 0.9180). Dan Wei [7] in his research paper also favors the use of SVM as it gives a fairly accurate F score of 0.839 over his testing data. Other papers [8] suggested that there is 1.5 times more research devoted to estimation modeling than forecast modeling. Estimation-based experiments predominantly use ensemble learning and regression algorithms, while forecasting activities prefer to use methods based on NN and SVM. A. Masih points out that there is a clear link between predictive features such as land use and satellite imagery with estimation models, but the correlation with forest models is quite poor. Ensemble learning is a highly accurate process with an average correlation coefficient which is close to 0.8, but there are only a few implementations for forecast modeling. In their study, Kunwar P.Singh and Shikha Gupta [11], to differentiate between seasonal air characteristics, factors responsible for segregation, and to forecast air quality indices, studied and developed various ensemble models. Accordingly, Single Decision Tree (SDT), Decision Tree Forest (DTF), and Decision Tree boost (DTB) were then built and their generalization and predictive efficiency were tested in terms of multiple statistical parameters and support vector machines (SVM) relative to traditional machine learning benchmarks. They identified that major sources of air pollution in the city of Lucknow (India) are emissions from vehicles and fuel combustion. They used Principle Learning Analysis (PCA) for the same. Recent studies point out that [12] latest techniques in stochastic data analysis for discovering a collection of few stochastic variables used as

input for artificial neural network models for air quality forecast, representing the related information on a multivariate stochastic system can be a turning point in AQI data processing. Ana Russo and fellow authors demonstrate that it is possible to substantially reduce the number of input variables required for the neural network model by using these derived variables as input variables for training the neural networks, without significantly modifying the model's predictive capacity. Work done in [13] doesn't forecast PM2.3, but it helps in determining other factors like temperature, wind, dew point. It uses a Deep Hybrid Model for the same. They also derive an efficient learning and inference procedure that allows for large scale optimization of the model parameters. The complexity of AQI forecasting is caused by the randomness, non-stationarity, and irregularity of the air quality index (AQI) set. Using a hybrid approach (a proposal) [15] which combined a regression model as well as a chemical transport model, authors were able to estimate pollutants including PM2.5 in spacetime continuous. The experimental findings of other papers [14] also indicate that because of its higher forecast precision, the proposed hybrid model based on a two-phase decomposition methodology is strikingly superior to all other models considered. In their paper [16], based on RMSE and MAE calculations, authors, Mahajan, Sachit & Chen, Ling-Jyh, were able to conclude that the NNAR based model gives more promising results and better prediction accuracy as and when in comparison to the Holt-Winters based model and ARIMA based model. The authors used the AirBox data to propose a model that was able to precisely forecast PM2.5 for the upcoming hour.

III. METHODS

The problem here is to automate the scheduling of an air purifier based on the data collected from the sensor, or from an actively updating website which provide us with the necessary data, (more about which is discussed below) and then passing through our model and predicting an accurate Air Quality Index and hence based on predicted Air Quality Index, passing command to purifier to toggle it between ON and OFF mode.

For this, our first step will be to get a good dataset. A good dataset must have a good number of data points. Data points act as a training point for a model so the more the number of data points in our dataset, the better for our model to train on and also more sample data to test it on and hence better and more accurate accuracy. For this specific problem, we need a dataset with the concentration of different harmful particles as our attributes (features) and an accurate Air Quality Index for that specific concentration as our target column. Here we have used the dataset available at https://www.kaggle.com/rohanrao/air-quality-data-in-india. One more thing one should consider while selecting a dataset is the number of missing values in that dataset (number of NULL or NaN values), this makes data less accurate in the first place as well as increase our work of data pre-processing.

Next comes data pre-processing. Data Pre-processing gets data Encoded, or transformed to convert it so that it now gets easily parsed by machine. In other words, a conversion which makes the features in the dataset easy to be interpreted by the model (algorithm). Data pre-processing helps us deal with the following problems in our dataset: 1. Missing values, 2. Inconsistent values, 3. Duplicate values. Here, for the dataset used in this problem, we have dropped missing values. Feature encoding, which is transforming data such that it is easily accepted by the machine learning algorithms and it still retains its original meaning. For Feature encoding, we have used MinMaxScaler. MinMaxScaler transforms features by scaling them to a given range (which by default is 0-1). Here we have transformed all our features (X) using MinMaxScaler. The last thing we do in data preprocessing is splitting the data into different sets which can be 2 (Train and Test data) or 3 splits (Train, test, and validation data). Machine Learning algorithms are trained on the training dataset and then is validated and tested on test and validation data before it is deployed to deal with real-world data (More the test set accuracy better the model and more the train set accuracy more our model is inclined towards overfitting). Here we have split our data into 2 sets with a split of 0.3 which means the train data will be 0.7 of our total dataset and test data will be 0.3 of the same.

After that comes model selection, which means selecting a model on which we will fit our data and train it to help it predict the real-world data. It consists of 2 steps, Model selection, and model evaluation. The selection of the model depends on the type of data we have, what features are there, how it is scaled, what needs to be predicted, and much more. After that comes its evaluation which may be bad due to data, we might have not chosen an optimal model for our data or another issue maybe with parameters and hyperparameter used which we have used in our model which may have made our model overfit or underfit. So, we need to select an optimal model by considering these points. Here we have chosen RandomForestClassifier as our model and based on the data we have set our parameters and hyperparameters accordingly like here for this problem we have set out ‘max_depth’ parameter as 9 and 250 trees which was giving us an optimal solution for our dataset, the optimal solution here means that the model is neither overfitting nor underfit and gives a good test accuracy. Here for selected parameters and model, we are getting a test

accuracy of 85% which is good enough given that the model is not inclined towards overfitting or underfitting. Next step here is fetching real time data to use to predict real time air quality index of that area using the model trained above. This is done using python's functions to scrap the web. The website which we are scraping data from is https://aqicn.org/city/ + the city's name where we are running our model to get data (concentration of harmful particles) of that specific region, which was the sole purpose of our project. This website has current data about concentration of particles as well as past data of the same, but we are only interested in current data so we have fetched only the current data for concentration. We have used pyhton's 'requests' package which has inbuilt functions to do so and after fetching our required data we have transformed it and stored in a similar way as our test set data so that it can be passed to our model to predict Air quality index which then send a signal to our purifier, toggling it between ON and OFF modes. Concentration of particles is not constant and it keeps changing so, we need to predict air quality index periodically. The period of this periodic run should be when there is significant change is the values of the concentration of particles, but we can't refresh the concentration every moment in real time as we are dependent on the website from where we are fetching our data and it gets updated every hour so we need our model to fetch new values of concentration every hour, run to through our model to predict Air Quality Index and also pass a corresponding single to our purifier. So this needs to be updated and hence the last step.

The last step here will be its automation. For this problem as we discussed, we are scraping data from a website so, we need it to run again for any changes in the concentration of particles to predict real-time Air Quality Index and hence toggle our Air purifier accordingly. The webpage from where we have scrapped our data https://aqicn.org/city updates its city data for a concentration of harmful particles every hour, so we need to make our web scrapping code and prediction code run again every hour when the webpage gets updated so that we can toggle our air purifier on the most recent real-time data available. For this, we have used python's library 'apscheduler' and from the 'BlockingScheduler'. BlockingScheduler run a function every given interval, the function defined to run every interval is 'now', 'now' function calls two more functions, first being the function named 'getCurrent' which scraps data from the website to be then passed to model for prediction, this is done by 'model.predict()' this is a built-in function which takes input as the data on which we need to predict the value of bypassing the data to the model, it returns the value which there is Air Quality Index which is then processed to send a signal to purifier to toggle between modes based on the predicted value from the model.

IV. RESULTS:

Working on this dataset we have obtained some really interesting visuals regarding the air quality of India.

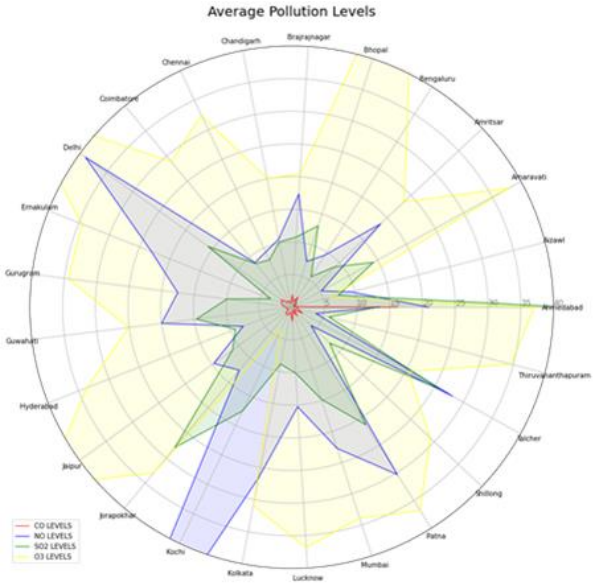

Fig1. Concentration of different pollutants across India

For the purpose of classifying the data, we have used a number of models and each of their accuracy is listed below:

Table 1. Test Accuracy of different models

S.No.	***Model Name***	***Accuracy on Test Data (In percent)***
1.	KNeighbors classifier	71.56
2.	MLP Classifier	70.88
3.	Random forest classifier	80.02
4.	Decision tree classifier	65.35
5.	Support Vector Machine	68.04
6.	Bagging Classifier	73.34
7.	Svm linear svc (max_iter= 1000)	56.28

As it can clearly be observed from the above table that we have achieved maximum accuracy when we used Random forest classifier, so we chose that model and after some hyperparameter adjustments we were able to attain an accuracy of 81.13%.

Below are some working screenshots:

1. After using the random forest classifier, this is the testing and training accuracy that we achieved.

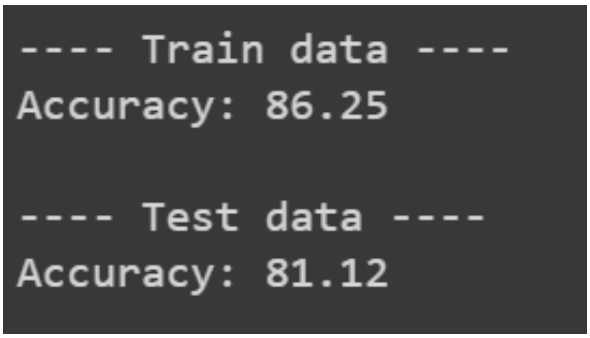

Fig2. Accuracy of our model

2. Getting data from https://aqicn.org/delhi

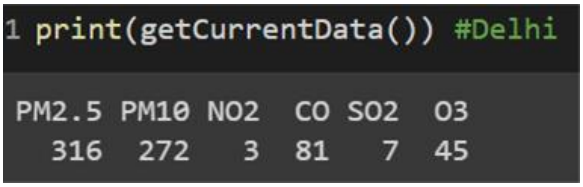

Fig3. WebScraped AQI of Delhi

3. Model from fig2. and data from fig3.

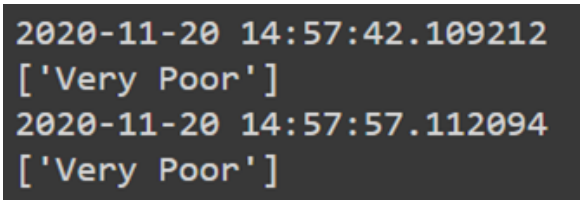

Fig4. Predicted AQI Bucket at an interval of 15 seconds

V. DISCUSSION

Theoretical and realistic importance for reliable air quality forecasting is essential for the people; without it, neither the government nor the public can efficiently prevent the health harm caused by air pollution or enhance the ability of high pollution days to respond to emergencies. In this research, we constructed a random forest model based on machine learning algorithms to forecast air indicators. The data set used is based on the AQI (Air Quality Index) predictions and other relevant gases of different regions in India. Running our model on Real-time AQI values of specific regions gave us results that helped us to summarize that this model can achieve promising results and with a larger and more complex training data set there's room for improvement in the accuracy as well.

VI. REFERENCES:

[1] Liu, Bing-Chun & Binaykia, Arihant & Chang, Pei-Chann & Tiwari, Manoj & Tsao, Cheng-Chin. (2017). Urban air quality forecasting based on multi-dimensional collaborative Support Vector Regression (SVR): A case study of Beijing-Tianjin-Shijiazhuang. PLoS ONE. 12. 10.1371/journal.pone.0179763.

[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] Allen, R.; Larson, T.; Sheppard, L.; Wallace, L.; Liu, L.J.S. Use of Real-Time Light Scattering Data to Estimate the Contribution of Infiltrated and Indoor-Generated Particles to Indoor Air. Environ. Sci. Technol. 2003, 37, 3484–3492. [CrossRef] [PubMed]

[4]Kabir S.i ; Raihan U.I., Hossain S.M. , Andersson K.An (2020 March 5) Integrated Approach of Belief Rule Base and Deep Learning to Predict Air Pollution

[5] Ahn, J., Shin, D., Kim, K., & Yang, J. (2017, October 28). Indoor Air Quality Analysis Using Deep Learning with Sensor Data. Retrieved September 23, 2020, from https://www.ncbi.nlm.nih.gov/pubmed/29143797

[6] Liu H., Qing L., Dongbing Y., Yu Gu. (2019 September) Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms

[7] Wei, D. (2014). Predicting air pollution level in a specific city.

[8] A. Masih.(17 July 2019). Machine learning algorithms in air quality modeling

[9] Di Nicolantonio, Walter & Cacciari, Alessandra & Tomasi, Claudio. (2010). Particulate Matter at Surface: Northern Italy Monitoring Based on Satellite Remote Sensing, Meteorological Fields, and in-situ Samplings. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of. 2. 284 - 292. 10.1109/JSTARS.2009.2033948. .

[10] Bellinger, Colin & Zaïane, Osmar & Osornio Vargas, Alvaro & Jabbar, Mohomed. (2017). A systematic review of data mining and machine learning for air pollution epidemiology. BMC Public Health. 17. 10.1186/s12889-017-4914-3.

[11] Singh, K., Gupta, S., & Rai, P. (2013, August 22). Identifying pollution sources and predicting urban air quality using ensemble learning methods. Retrieved September 23, 2020, from https://www.sciencedirect.com/science/article/pii/S1352231013006328

[12] Russo A., Frank R., Pedro G.Lind (2012) Air quality prediction using optimal neural networks with stochastic variables

[13] Aditya Grover, Ashish Kapoor, and Eric Horvitz. (2015). A Deep Hybrid Model for Weather Forecasting. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). Association for Computing Machinery, New York, NY, USA, 379–386. DOI:https://doi.org/10.1145/2783258.2783275

[14] Wang D., Wei S., Luo H., Yue C., Grunder O. (2017 February 15) A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine

[15] Di, Qian & Schwartz, Joel & Koutrakis, Petros. (2016). A Hybrid Prediction Model for PM2.5 Mass and Components Using a Chemical Transport Model and Land Use Regression. Atmospheric Environment. 131. 10.1016/j.atmosenv.2016.02.002

[16] Mahajan, Sachit & Chen, Ling-Jyh & Tsai, Tzu-Chieh. (2017). An Empirical Study of PM2.5 Forecasting Using Neural Network. 10.1109/UIC-ATC.2017.8397443.