IoT and ML-based AQI Estimation using Real-time Traffic Data

Nitin Nilesh, Ayu Parmar, Ritik Yelekar

Abstract

This report proposes an IoT and machine learning (ML)-based novel method to estimate the air quality index (AQI) using traffic data in real-time. With the help of particulate matter (PM) monitoring nodes deployed in fifteen locations with diverse traffic scenarios of Indian roads, and using digital map service providers, a rich traffic dataset with approximately 210,000 samples has been collected. Three different ML models, namely random forest (RF), support vector machine (SVM), and multi-layer perceptron (MLP), are trained on this dataset to predict the AQI category into five levels. The experimental results show an accuracy of 82.60% with the F1-score of 83.67% on the complete dataset. Apart from this, ML models were also trained on individual node datasets, and the behavior of AQI levels was observed.

1 Introduction

Air pollution is a grave threat to human health and the environment. As a result of long-term exposure to air pollution, millions of people die every year while several more get seriously ill [1]. Therefore, monitoring air pollution is essential to address the threat. For this, Governmental agencies (such as Central Pollution Control Board (CPCB) in India) deploy scientific-grade devices to monitor air quality. Although the air quality data from these stations is very accurate, this approach has the limitation of scalability as these devices are extremely costly, bulky and difficult to maintain [2].

With the rise of internet-of-things (IoT) based low cost air pollution monitors, dense deployments for improving spatial resolution of air quality data have been possible in recent times [2,3]. However, the sensors-based approach still has several issues. For example, low-cost sensors have low-accuracy, need calibration seasonally, and have limited lifetime (few months or year). For example, one of the popular particulate matter (PM) sensor SDS011 by Nova Fitness [4] has lifetime of 8000 hours (approximately one year). For this reason, it is desirable to have a method which does not depend on sensors. In this project, we propose such a method using real-time traffic data.

2 Goals

The specific aim and goal of the project are as follows:

- An IoT and ML-based methodology is proposed to estimate the real-time AQI into five levels using real-time traffic data and weather parameters. To the best of the authors' knowledge, this article is the first of its kind to achieve this on Indian roads.
- A completely new rich traffic dataset has been collected containing approximately 210,000 data points, including traffic information (such as the mobility rate of the traffic), weather information (temperature and relative humidity) and co-located ground truth PM values. The dataset contains samples across the 15 different locations in Hyderabad from Jan'22 - May'22.
- A simple yet effective ML algorithm is used to estimate the AQI level, which enables the whole pipeline to be fast and real-time with minimal processing.
- The proposed method achieved an overall accuracy of 82.60% with an F1-Score of 83.67%. We also show the results on individual traffic locations to better understand the scenario.

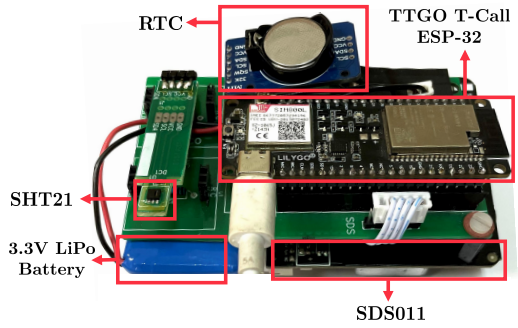

Figure 1: PM monitoring device [3].

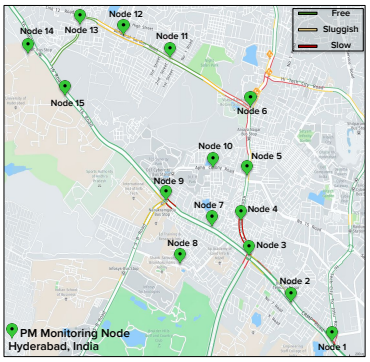

Figure 2: Locations of the 15 PM monitoring nodes on the map along with the traffic status.

3 System Architecture and Design

3.1 Hardware

Table 1: Specifications of sensors used in the developed PM monitoring node.

Sensors	Specification	Value
SDS011 [4]	Measurement parameters	$PM_{2.5}$ & PM_{10}
	Operating Temp Range	-20°C to +50°C
	Operating RH Range	0-70%
SHT21 [5]	Measurement parameters	Temp & RH
	Operating Temp Range	-40°C to +125°C
	Operating RH Range	0% RH to 100% RH

Fig. 1 show the block the circuit board, of the IoT PM monitoring device deployed in the main road and junctions. Each node consists of TTGO T-Call ESP32 [6] based microcontroller and sensors for PM, temperature and humidity; additionally, it has a real-time clock (RTC) and Li-Po battery. The sensors' specifications are given in Table 1. Nova PM SDS011, which is a light scattering principle-based sensor, has been used for measuring the concentration of fine particulate matter $PM_{2.5}$ and PM_{10}, as it has been shown to have the best performance with beta attenuation mass (BAM) compared to other low-cost sensors [7]. As concluded in reference, that temperature (Temp) and relative humidity (RH) impact PM concentration and also the light scattering-based PM sensors do not perform reliably well at extreme temperature and humidity conditions. SHT21 is used to monitor these parameters for the reliability of SDS011 sensor readings.

The controller reads data from all the sensors periodically at a frequency of 30 sec and offloads it to ThingSpeak, a cloud-based server employing MQTTS. The SDS011 and SIM800L modules are connected to the controller through the UART protocol, while the SHT21 and RTC are connected through the I2C protocol. The device is powered using a 3.3V rechargeable lithium polymer ion battery. An AC-to-DC Power adapter and an onboard battery management circuit are used to charge the battery. As the deployment is outdoor, the sensor node is enclosed in a polycarbonate box of IP65 rating, which protects the node from dust and water.

Fig. 2 shows the location of the nodes and the traffic status of the roads (on an average day). These locations

are used to collect the real-time traffic data as well as sensor data. These locations mainly contain major city roads and include a mixture of heavy and light traffic. The total distance covered is approximately 15 kms spanning an area of 6 km^2.

3.2 Dataset Collection

In this paper, a dataset is collected using the PM monitoring node defined in the section **??**, with the help of digital map service providers. A 5-dimensional feature vector has been accumulated for each data point in the dataset, where the features are

- Traffic mobility rate (TMR)
- Humidity
- Temperature
- Normalized difference vegetation index (NDVI)
- Time of the day (categorized as morning, afternoon and evening)

After concatenating all the features accumulated from the samples in the dataset, a $m \times 5$ data matrix M is obtained, where m is the number of samples present in the dataset. A $m \times 1$ sized vector y containing the corresponding label for each sample is the respective AQI category computed using $PM_{2.5}$ and PM_{10} values. Next, some of the important parameters such as TMR, NDVI, and AQI categorization are explained in more detail.

3.2.1 Traffic Mobility Rate

Traffic on the road is defined as the rate of mobility of the vehicles present on the road. In standard speaking terms, high traffic refers to the slow mobility of the vehicles and vice versa. There are several ways to get the traffic status of a specific location (road) using various digital map service providers, e.g., Google Maps, HERE Maps, Bing Maps, etc. An application programming interface (API) from HERE Maps [8] is used for our use case to collect the traffic data in real-time. HERE Maps RESTful web API provides location-aware features such as traffic and weather information. For a given location (latitude and longitude) with the desired radius, HERE Maps API returns a list of roadways and their traffic information in real-time.

One of the vital traffic information provided by HERE Maps API is the mobility score of a given road, also known as the Jamming Factor (JF). The JF is a real number between 1 and 10 and is categorized as follows: 1. **Free** traffic flow (0 - 4) 2. **Sluggish** traffic flow (4 - 8) 3. **Slow** traffic flow (8 - 10). As for a given location, there are multiple roadways and each roadway has a JF associated with it, we calculate the final traffic mobility rate (TMR) as follows:

$$T = \frac{1}{n} \sum_{r=1}^{n} J_r \tag{1}$$

where J_r is the JF of the r^{th} road and n is total number of roads for any given location.

The traffic parameters are collected every 30 seconds between 0800 hrs and 2100 hrs across the month of Jan'22-May'22. A total of approximately 210,000 samples have been collected in this duration. One of the significant reasons to collect the data in the daytime is to predict the behavior of air quality only in the presence of traffic, as in the night-time, the traffic is negligible.

3.2.2 NDVI Score

The NDVI [9] is a graphical indicator that indicates the presence of vegetation in a particular area. It is a technique to classify the land as green, barren, etc., using satellite images of the earth. [10] shows that vegetation is a sound-absorbent of PM. It helps settle the dust and acts as a natural bio-filter against the PM.

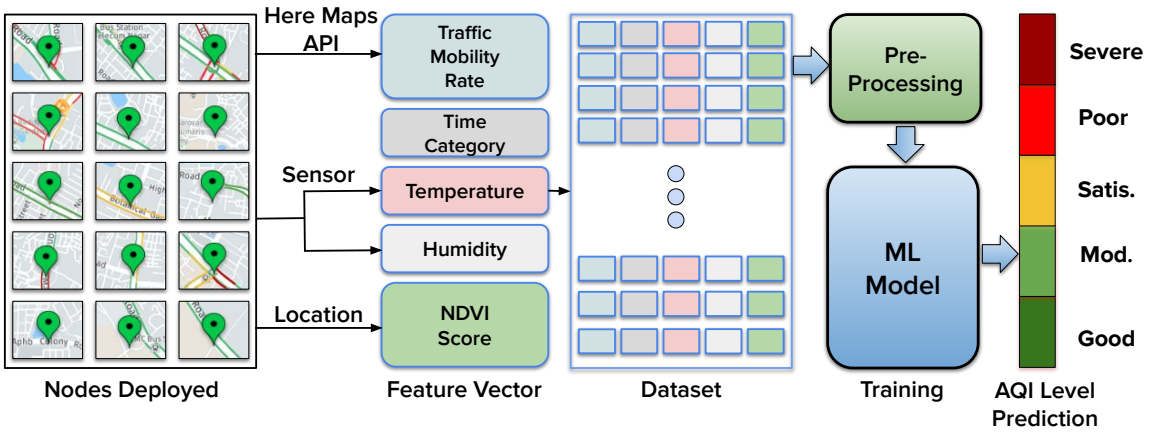

Figure 3: Algorithmic pipeline for the proposed methodology. TMR has been calculated using HERE Maps API using the nodes deployed and their location. After that, the NDVI score and weather information (temperature and humidity) are concatenated to make a 5-dimensional feature vector dataset. Further, this dataset is used to train the ML model to predict the AQI level. (Best viewed in color)

In our case, we try to locate the vegetation areas for the given map in Fig. 2 and then relate the PM values captured by the sensor. As the vegetation in an area can affect the AQI category, it is essential to consider this score while predicting the AQI. NDVI score is calculated as follows:

$$\text{NDVI} = \frac{R_{nir} - A_{red}}{R_{nir} + A_{red}} \tag{2}$$

where R_{nir} is the amount of reflection on the vegetation area in the near-infrared spectrum and A_{red} is the amount of absorption onto the vegetation area in the red range of the spectrum. The NDVI value ranges between -1 to $+1$, where -1 indicates a high probability of water body and $+1$ indicates a high probability of vegetation in that area. For our paper, the NDVI values for the 15 locations were collected every month as the change in the vegetation is very slow.

3.2.3 AQI Categorization

Each sampled data point of the dataset is associated with co-located respective node sensor values, i.e., temperature, relative humidity, $PM_{2.5}$, and PM_{10} measurement. The AQI level is computed using the $PM_{2.5}$ and PM_{10} values as per the Central Pollution Control Board, India [11], and categorized into five classes which are as follows: 1. **Good** (0 - 50) 2. **Satisfactory** (51-100) 3. **Moderate** (101-200) 4. **Poor** (201-300), and 5. **Severe** ($>$300).

3.3 Software

The main idea of this work is to predict the AQI for a given traffic scenario. We propose a simple yet effective methodology to predict the AQI category using the dataset defined in section 3.2. The pipeline shown in Fig. 3 explains the proposed methodology. Firstly, the dataset is preprocessed, and then used to train the ML model. Further, the trained ML model is used to predict the AQI category for a given test sample.

3.4 Preprocessing

Before training the ML model, the first step is to preprocess the dataset M obtained in section 3.2 so that it follows characteristics helping better model generalization. Standard normalization is applied to the dataset for data preprocessing to achieve zero mean and unit standard deviation. This step ensures that all the

samples in the dataset follow a similar data distribution and helps converge faster while training the model. After this, a *MinMax* scaler is applied to the dataset, transforming all the features into a range of 0 to 1. This step ensures that all the features of the dataset are in the same range avoiding any kind of bias in the model. The whole preprocessing step is defined as follows:

$$M_s = \frac{M - \mu_M}{\sigma_M} \quad \text{(Standard normalization)}, \tag{3}$$

$$M' = \frac{M_s - \min(M_s)}{\max(M_s) - \min(M_s)} \quad \text{(\textit{MinMax} scaling)} \tag{4}$$

where μ_M and σ_M is the mean and standard deviation of M along the columns respectively.

3.5 Training

With the help of preprocessed dataset, M' defined in the above section, and the corresponding label vector y, a ML model was trained to classify the samples into five different AQI categories. As this is a supervised learning problem, a classification-based ML model was used. The dataset's features M' contain both kinds of values, i.e., continuous and discrete. All the values in the dataset are well normalized with a similar range of values. Due to these factors, we chose ML models that best suit the dataset. We experimented with three different ML models: 1. Random Forest (RF) [12], 2. Support Vector Machine (SVM) [13] and, 3. Multi-Layer Perceptron (MLP) [14] and choose the best performing model after hyperparameter tuning. Each model's output was set to five classes depicting the respective AQI categories.

While training, the training dataset M' was split using the K-fold cross-validation technique with ten folds. The ML model was trained and validated on each fold separately. This is a paradigm used while training the ML models to increase the model's generalization ability. During the evaluation of the ML model, four metrics were calculated: 1. Accuracy, 2. Precision 3. Recall, and 4. F1-score on the validation part and mean was taken across all ten folds.

3.6 Detection

At the time of detection, firstly the traffic mobility rate is fetched using the HERE Maps API for a given location. After that, the NDVI score for that particular area is obtained. These values are concatenated with the humidity and temperature of the location along with the time of the day, making a feature vector of size 5×1. This feature vector was first preprocessed using the methods defined is subsection 3.4. After this, the trained model was used to predict the AQI category into one of the five classes.

4 Addressing Challenges

The key challenges encountered while building the system:

- Deployment of 15 nodes on traffic junction poles.
- Data collection and sensor failing.

Deployment and data collection were the most challenging task, as devices were failing. To overcome these challenges, we tried to make the device more rigid and worked on the network connectivity part.

5 Performance Evaluation and Testing Results

As discussed in the proposed methodology (section **??**), a total of three models were experimented and trained to classify the AQI on the dataset defined in section 3.2. For the RF model, the number of decision

trees was set to 200 as the number of samples in the dataset is large, and the split criterion was *entropy*. Tree pruning mechanisms were used to avoid overfitting and get better convergence while training the RF model. For the SVM model, the *regularization* parameter(*C*) was set to 8 with *Radial Basis Function* kernel. In the case of the MLP model, five hidden layers with neuron sizes 128, 64, 32, 16, and 8, respectively, with *Rectified Linear Unit* (ReLU) non-linear activation function, was used to train for 100 epochs. All these models were implemented using *Scikit-Learn* [15], which is a popular python-based ML library. As these ML models do not have many parameters to train, they are computationally very light and took only a few microseconds while inferencing on the single test sample.

Table 2 shows the importance of the features in the dataset while computing the AQI. It can be observed that the feature traffic mobility rate and NDVI score play an essential role with the support of temperature and rest other features.

Table 2: Importance of features w.r.t AQI

TMR	NDVI	Temperature	Humidity	Time of the Day
0.32	0.29	0.19	0.11	0.09

Table 3: Performance of the various ML models on overall dataset.

ML Model	Accuracy	Precision	Recall	F1-Score
RF	**82.60%**	**84.73%**	**82.63%**	**83.67%**
MLP	79.31%	77.98%	79.43%	78.70%
SVM	78.52%	77.13%	78.67%	77.89%

The ML models were trained and validated for the dataset mentioned in section 3.2. The environmental factor around them is diverse as the dataset is collected on a total of 15 different nodes at different locations. Due to this reason, two types of ML models were trained: 1. ML model on the overall dataset, and 2. Individual ML models for each node's dataset. Each model's performance was evaluated on four different metrics named accuracy, precision, recall, and F1-score. The results obtained for the overall and individual nodes are reported in table 3 and 4 respectively. For the overall dataset, the RF model performed the best with an accuracy of 82.6% and an F1-Score of 83.67%. In the case of the individual dataset, it can be observed from Fig 2 that Node 6, 8, and 11 are near high vegetation areas. For these nodes, the AQI level for most of the data points fell in the first two categories, i.e., "Good" and "Satisfactory". Hence, the model's task was easier for these nodes and performed better than the rest of the node's data.

On the other hand, for Node 1 and 3, the traffic mobility rate is high as they are placed at road junctions. For these nodes, the TMR varied mainly from "Sluggish" to "Slow", resulting in Poor to Moderate AQI levels with some instances of Severe as well.

6 Concluding Remarks and Avenues for Future Work

This article introduced an IoT-based technique to predict the AQI from traffic and location data in real-time. Location-based features like traffic mobility rate, NDVI score, and sensor-based features like temperature and relative humidity were used to train the ML model. Additionally, a dataset having around 210,000 samples that contain traffic and weather information is collected and to be released in the public domain to promote further research. Experimental results show an F1-Score of 83.67% for the overall dataset, while

Table 4: Performance of the ML model on individual node's dataset. Please note that the best performing ML model result is shown.

# Node	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	79.69	78.16	79.93	79.03
2	79.56	78.11	81.03	79.54
3	78.21	79.55	78.26	78.90
4	78.51	79.36	79.61	79.48
5	79.86	78.75	78.42	78.58
6	82.78	84.70	81.72	83.18
7	78.95	79.90	79.3	79.59
8	80.15	82.42	81.98	82.20
9	79.56	79.96	79.90	79.92
10	78.98	82.49	80.29	81.37
11	81.67	84.60	81.26	82.89
12	82.94	84.31	80.82	82.52
13	79.88	84.78	79.30	81.94
14	81.05	83.44	80.37	81.87
15	82.63	79.96	78.65	79.29

experiments on node-specific datasets show the sensitivity of the location. ML model performance on locations having high vegetation index performs better than others, specifically where the vegetation is low, and traffic is peak.

7 Availability

- ML model source code: https://github.com/ayuparmar/Here_Maps
- Video: https://iiitaphyd-my.sharepoint.com/:v:/g/personal/ayu_parmar_research_iiit_ac_in/EcdevEBhARJPpELooy37BXQBVR_kSN5v3o87FhsTFQxcIA?e=B7Qo2z

References

[1] P. Landrigan et al., "The lancet commission on pollution and health," *Lancet*, vol. 391, pp. 464–512, 2018.

[2] C. Rajashekar et al., "Improving spatio-temporal understanding of particulate matter using low-cost iot sensors," in *Int. Symp. Personal, Indoor and Mobile Radio Commun.*, 2020.

[3] A. Parmar et al., "Development of low-cost IoT device for densely deployed PM monitoring network: An indian case study," to be submitted in IEEE Internet of Things Journal, https://bit.ly/3xGsmG8.

[4] *SDS011 Nova Sensor Specifications*, accessed 2021, http://www.inovafitness.com/en/a/chanpinzhongxin/95.html.

[5] *SHT21 Specification*, accessed 2021, http://www.farnell.com/datasheets/1780639.pdf.

[6] *TTGO T-Call ESP32 module Specifications*, accessed 2021, https://docs.ai-thinker.com/_media/esp32/docs/esp32-sl_specification.pdf.

[7] M. Badura, P. Batog, A. Drzeniecka, and P. Modzel, "Evaluation of Low-Cost Sensors for Ambient PM2.5 Monitoring," *Journal of Sensors*, vol. 2018, no. 5096540, pp. 1–16, 2018.

[8] *HERE Maps Traffic API*, accessed 11 June. 2022, https://developer.here.com/documentation/traffic/dev_guide/topics/what-is.html/.

[9] J. Weier and D. Herring, "Measuring vegetation (ndvi & evi)," *NASA Earth Observatory*, vol. 20, p. 2, 2000.

[10] A. Diener and P. Mudu, "How can vegetation protect us from air pollution? a critical review on green spaces' mitigation abilities for air-borne particles from a public health perspective - with implications for urban planning," *Science of The Total Environment*, vol. 796, p. 148605, 2021.

[11] *National Air Quality Index*, accessed 11 Apr. 2022, https://app.cpcbccr.com/AQI_India/.

[12] L. Breiman, "Random forests," *Machine learning 45.1 (2001): 5-32*, 2001.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[15] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, 2011.