Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 241–248

Procedia Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 6 - 9, 2020, Warsaw, Poland

Air Quality Forecasting using LSTM RNN and Wireless Sensor Networks

Sagar V Belavadi[a,*], Sreenidhi Rajagopal[a], Ranjani R[a], Rajasekar Mohan[a]

[a]*Department of Electronics and Communication Engineering, PES University, Bangalore, 560085, India*

Abstract

In the past few decades, many urban areas around the world have suffered from severe air pollution and the health hazards that come with it, making gathering real-time air quality and air quality forecasting very important to take preventive and corrective measures. This paper proposes a scalable architecture to monitor and gather real-time air pollutant concentration data from various places and to use this data to forecast future air pollutant concentrations. Two sources are used to collect air quality data. The first being a wireless sensor network that gathers and sends pollutant concentrations to a server, with its sensor nodes deployed in various locations in Bengaluru city in South India. The second source is the real-time air quality data gathered and made available by the Government of India as a part of its Open Data initiative. Both sources provide average concentrations of various air pollutants on an hourly basis. Due to its proven track record of success with time-series data, a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) model was chosen to perform the task of air quality forecasting. This paper critically analyses the performance of the model in two regions that exhibit a significant difference in temporal variations in air quality. As these variations increase, the model suffers performance degradation necessitating adaptive modelling.

Keywords: Air Quality Forecasting; Wireless Sensor Network; Sensor Node; Long Short-Term Memory; Recurrent Neural Network.

1. Introduction

Air quality refers to the concentration of various air pollutants in the atmosphere at a given time and place [1]. Air pollutants include gases such as Sulphur Dioxide (SO_2), Ammonia (NH_3), Nitrogen Dioxide (NO_2), Carbon Dioxide (CO_2), Carbon Monoxide (CO), Ozone (O_3) and particulate matter (PM 2.5 and PM 10). Various studies show that short-term exposure to elevated levels of these pollutants may cause difficulty breathing, eye irritation and may lead to pulmonary and cardiovascular health effects. Long-Term exposure may lead to cancer and damage to the respiratory,

* Corresponding author. Tel.: +91-8762733652.
E-mail address: sagar.v.belavadi@gmail.com

10.1016/j.procs.2020.03.036

reproductive, neurological and immune systems of the body. This makes air quality forecasting and gathering real-time air quality data an important area of research.

The measurement of air pollutant concentrations to a high degree of accuracy has required the use of expensive sensors which are difficult to deploy and operate. This has stifled community participation and has led to a poor spatial density of gathered air quality data. In this paper, an architecture to gather real-time air quality data is proposed that consists of a Wireless Sensor Network (WSN) made up of low-cost sensor nodes for monitoring and transmitting air pollutant concentrations. The sensor nodes contain various easily available low-cost sensors for measuring air pollutant concentrations, a communication module for transmitting the measured data and a microcontroller for coordinating these tasks. Although the data gathered by these nodes tend to be noisier, it can nevertheless be used by machine learning and deep learning algorithms to extract useful trends and make predictions.

The task of air quality forecasting is that of predicting the concentration of various air pollutants based on the current concentration of the pollutants recorded by sensors in real-time and trends in historical data. Since the historical data here is time-series data of recorded pollutant concentrations, a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) model was chosen for the task due to its proven track record with time-series data[2]. This paper makes a novel contribution by exploring LSTM RNN which is a deep learning algorithm for the task of air quality forecasting for which only machine learning algorithms have been explored[10]. Real-time air quality data gathered and made available by the Government of India as a part of its Open Data initiative [3] was used to form the historical data which was used to train the LSTM RNN models.

To make real-time air quality forecasts, a web application was developed. The application queries the data sources and saves the received the data into a database. Additionally, it also feeds this data to the stored pre-trained LSTM RNN models which in turn utilise these feeds to forecast air quality for the succeeding five hours. The web application also serves a front end web page for users to select and view the pollutant data and forecasts. The focus of this paper will be to study the performance of the LSTM RNN models for the task of air quality forecasting and to design the low-cost sensor nodes that form a WSN for monitoring air quality.

2. Literature Survey

Wireless sensor networks have been widely implemented in many applications ranging from industrial monitoring and control, home automation, medical disaster response and many more which are discussed in depth in [4]. Some examples of WSN's to detect real-time environmental conditions are explored in [5] and [6]. In [5], sensors are employed to gather environment data such as temperature, humidity, and soil moisture content which are utilised to implement a smart solar-powered irrigation system. The collected data is used to predict environment conditions using the Radial Basis Function Network (RBFN), which controls the irrigation system. A multi-sink distributed power control algorithm is proposed in [6] which exhibits superior connectivity, power consumption validity, and network performance of the WSN's in coal mines. It ensures that the monitored data is transmitted to the monitoring centre rapidly and quickly. A WSN model for air pollution monitoring in real time is proposed in [7]. The sensor stations form a distributed network which communicate to the back end server using machine to machine communication. Atmega 2560 microprocessor equipped with GPRS module is used to perform functions such as data acquiring, processing, logging, and transmitting.

Selection of the correct components is an important factor to consider while designing a WSN. In [9], hardware and software components of a typical WSN are listed, and analysis and criteria for their selection is discussed.

In [10], air quality forecasting by using machine learning approaches that predict the concentration of air pollutants in an hourly manner in the USA is proposed. We extend this by using LSTM RNN architecture, which perform very well on sequential data. LSTM theory is explored in detail in [11].

3. Methodology

3.1. Overview

A high-level view of the methodology is described in this section. The overall flow of the methodology is given in Fig. 1.

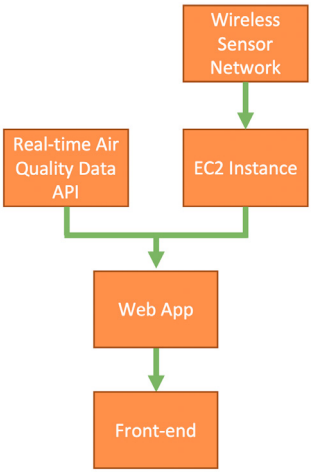

Fig. 1. Overall Methodology Flowchart

The role played by each part is as follows:

- **Wireless Sensor Network (WSN)**: The WSN consists of a network of sensor nodes deployed across Bengaluru, that collect and send real-time pollutant concentration data to the EC2 instance, with the time interval of one minute.
- **EC2 Instance**: It is a virtual server with a static IP address and Fully Qualified Domain Name (FQDN) running on Amazon's Elastic Compute Cloud (EC2). It listens for the data coming from the sensor nodes and saves it on the server. At the end of each hour, it averages the data and finds the minimum and maximum of the received data and makes it available through an API. It also provides a console that displays and plots the data being sent by a given node in real-time.
- **Real-time Air Quality Data API**: The central and various state Pollution Control Boards in India gather real-time data of various pollutant concentrations from various cities. This real-time data is made available hourly on data.gov.in and are accessible through a Data API [3].
- **Web App**: The web app is a Python web application, hosted using Google App Engine. The web app brings all the elements of the system together and accomplishes the following jobs:

    1. **Scraping**: The web app queries the Real-time Air Quality Data API and the EC2 instance twice every hour and saves the received data onto the database.
    2. **Forecasting**: A Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) model is used for forecasting a given pollutant's concentration. These models are trained beforehand on premises and are stored on the web app in the cloud. These stored pre-trained models are what constitute the forecasting engine which predicts the pollutant concentrations and will be explained in detail later. The web app feeds the new data it gets from the Data API and EC2 instance to the forecasting engine which then uses this to make forecasts for pollutant concentrations for the next five hours.
    3. **Serving**: The web app serves the front-end web page to the users' browsers and supports APIs used by the front-end to obtain the data.

- **Front-end**: The front end is the web page for users to select and view the pollutant data and forecasts. The users are required to choose which pollutant and monitoring station they want to view the data for and which date. The web page queries the API supported by the web app to get the required data and displays and plots it.

3.2. Wireless Sensor Network (WSN)

The WSN consists of a network of sensor nodes. Each sensor node is made up of the following components:

1. Sensors
   (a) MQ-2 Gas Sensor: To measure Carbon Monoxide (CO), Liquefied Petroleum Gas (LPG), and smoke.
   (b) MQ-135 Gas Sensor: To measure a mixture of gases such as Oxides of Nitrogen, Ammonia, Sulphide gases and Carbon Dioxide.
   (c) Sharp GP2Y1014AU0F Dust Sensor: To measure particulate matter in air (PM 2.5).
2. GSM Module: To establish a wireless connection to the EC2 instance and send the data to it.
3. Battery: To power the sensor node.
4. Arduino Uno microcontroller: To control and coordinate the above tasks.

3.3. Forecasting Model

The forecasting model is the Deep Neural Network responsible for forecasting the pollutant concentrations. An LSTM RNN architecture is used for the forecasting model. The process of building the forecasting model is as follows:

- The dataset is built by scraping the real-time air quality data API.
- Outlier removal is performed on the dataset.
- The dataset contains missing data points which reduce the efficiency of the forecasting model. Thus, to deal with these missing data points, we perform an imputation process.
- The result of the imputation is used to train the forecasting model.

Each of these steps is described in more detail in the next few sections.

3.4. Building The Dataset

The Government of India monitors various pollutant levels in many cities across India through over 160 monitoring stations. The pollutants monitored are Particulate Matter (PM10 and PM2.5), Sulphur Dioxide (SO_2), Nitrogen Dioxide (NO_2), Ozone (O_3) and Carbon Monoxide (CO). The average pollutant concentrations over an hour are updated hourly on data.gov.in and are accessible through a Data API [3]. This Data API is used by the scraper to build the dataset. The scraper queries the Data API and saves the returned data in a CSV file with a timestamp indicating the hour for which the data was collected. The scraper queries the Data API every 30 minutes and does not save the data received if it has not changed since the previous query. The scraper raises an alarm if there was an error in querying the Data API.

Thus, the scraper builds a list of CSV files each containing the minimum, maximum and average pollutant concentration for the given hour for all the pollutants monitored at each of the monitoring stations. This is restructured to generate our dataset which consists of one CSV file per monitoring station, pollutant pair which consists of two columns, the average concentration of the pollutant observed at that monitoring station and the time at which it was observed. Thus, it is a time series of the average pollutant concentration observed at a given monitoring station. This time series is used for training the forecasting model after some preprocessing. The data obtained after this step is shown in Fig. 2.

3.5. Outlier Removal

Outliers are data points that are distant from other data points. The training process of the model is very sensitive to the range and distribution of the data points. Outliers can be very misleading during the training process and result in longer training times and a less accurate model. Thus, the outliers are removed.

Z-scores have been used to identify if a given sample is an outlier. The z-score of a sample is the number of standard deviations it is away from its mean. Z-score of a sample is given by the formula.

$$z_i = \frac{(x_i - \mu)}{\sigma}$$

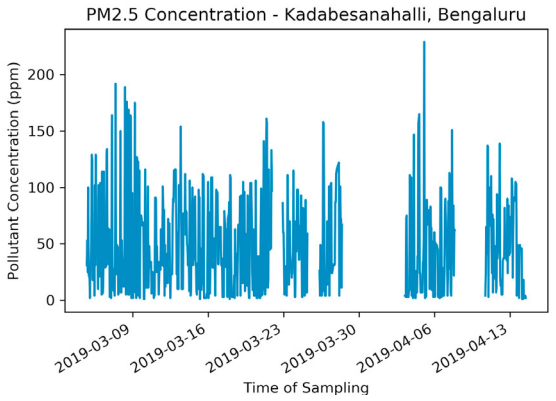

Fig. 2. Sample Data Obtained After Scraping

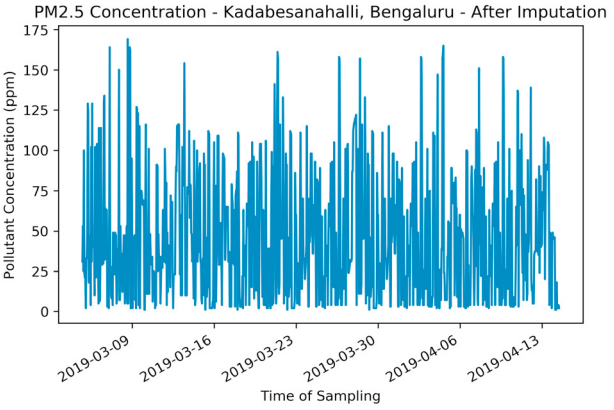

Fig. 3. Sample Data Obtained After Outlier Removal and Imputation

Where z_i is the Z-score of the ith data point, x_i is the ith data point, μ is the mean and σ is the standard deviation of the data. If the absolute value of the z-score of any sample is greater than 3, it is considered as an outlier and is removed from the dataset.

3.6. Imputation

The dataset obtained contains missing values for many hours. These missing values occur when the Data API does not provide a pollutant concentration which may be due to a sensor malfunctioning or a network or power outage at the monitoring station or the scraper. These missing values, cause temporal discontinuity and hamper the training process as the training process expects the data to be continuous. Thus, to solve this issue, missing values are substituted with the value recorded at the same time one week prior since they belong to the same distribution. If nothing was recorded at the same time one week prior, the mean value of the data is substituted for the missing value. The data obtained after this step is shown in Fig. 3.

3.7. Model Used

A Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) [11][12] architecture is used for the forecasting model. LSTM units are used as the building blocks for the LSTM RNN layers. The key part of an LSTM unit is the cell state which stores the information held by the LSTM unit. The LSTM unit can add or remove information from its cell state, regulated by structures known as gates. The LSTM unit also contains a hidden state which preserves past information of the observed sequence. The LSTM unit consists of three gates: ”forget gate”, ”input gate” and ”output gate”. These gates have the following functions:

1. The forget gate decides what part of the cell state is to be deleted based on the forget gate weights, hidden state and the current input.
2. The input gate decides how much of a new input is to be added to the cell state based on the input gate weights, hidden state and current input.
3. The output gate decides how the cell state will impact the current output based on the output gate weights, hidden state and current input.

The weights of the forget, input and output gates are crucial in deciding what is deleted from the cell state, what is added to the cell state and how much the cell state influences the output of the LSTM unit respectively. These weights are learnt by the model. Thus, the model learns over time what part of the cell state should be forgotten when an input arrives, which input is important and must be remembered and how much of the output should depend on the cell state for a given input. This allows the LSTM RNN model to effectively capture long-range dependencies in their cell states and overcome the problem of vanishing gradient faced by traditional RNN architectures.

Thus, the LSTM RNN model is perfectly suited for the air quality forecasting problem as the data is a sequence with long-range dependencies. The exact model used, that was constructed and trained using Tensorflow, consists of 3 layers:

1. LSTM Layer - 64 LSTM Units - Activation: ReLU
2. LSTM Layer - 32 LSTM Units - Activation: ReLU
3. Dense Layer - 1 Neuron - Activation: Linear

3.8. Training Process

The data generated after imputation is used for training the model. 67% of this data was used for training the model and the rest for testing the model. The training process minimises the mean squared error between the predicted value and the actual value, i.e, mean squared error is the loss.

Adam optimiser is used for training. The network weights are updated in an iterative manner based on the training data in this optimisation algorithm. The learning rate is separately maintained for each network parameter and it is adapted as learning happens. The network was trained for 50 epochs with early stopping which stops the training process if the model begins to over-fit, i.e, train loss is decreasing but test loss begins to increase.

3.9. Prediction Process

The model once trained can be used for making predictions. Given an input pollutant concentration, the model predicts the pollutant concentration for the next hour. To forecast the pollutant concentrations for the next five hours as required by the web app, the prediction for the n-1th hour is fed as the input to get the prediction for the nth hour.

3.10. The Web Application

The web app queries the Data API and the EC2 instance to collect the Real-time air quality data from both the Real-time Air Quality Data API and the EC2 instance twice every hour. If the data has not been updated since the previous query, then it is discarded; otherwise the received data is stored onto the database and used by the Forecasting Engine to make predictions for the next five hours.

4. Results

4.1. Prediction Results

Table 1 shows the performance results of the models trained on data from Amaravati, a small town in India. Similarly, Table 2 shows the performance results of the models trained on data from Bengaluru. Root Mean Squared Error(RMSE) is computed in ppm. It is also expressed as a percentage of the maximum value of the pollutant gas that was recorded(in ppm) during the period of observation between 03-04-2019 and 14-04-2019. Lesser the RMSE, better is the performance of the model.

Table 1. Results For Models Trained On Amravati Data

Station Name	Metric	CO	NH3	NO2	OZONE	PM10	PM2.5	SO2
Secretariat, Amaravati - APPCB	RMSE	3.14	0.23	1.02	5.11	4.72	4.96	1.04
	% of Max	7.70	7.45	1.68	6.89	3.25	5.71	4.77

Fig Nos. 4 and 6 plot the pollutant concentration data gathered during the observation period. 67% of this data was used for training the LSTM RNN model and the rest for testing the model. The predictions made by the model are plotted in Fig Nos. 5 and 7.

Table 2. Results for Models Trained on Bengaluru Data

Station Name	Metric	CO	NH3	NO2	OZONE	PM10	PM2.5	SO2
Bapuji Nagar, Bengaluru - KSPCB	RMSE (ppm)	34.78	32.84	34.90	35.22	35.39	35.26	33.68
	% of Max	26.62	26.25	28.42	27.40	25.93	27.02	26.68
BTM Layout, Bengaluru - CPCB	RMSE (ppm)	35.44	-	33.66	34.12	-	36.39	33.54
	% of Max	27.14	-	23.70	24.89	-	26.42	24.06
BWSSB Kadabesanahalli, Bengaluru - CPCB	RMSE (ppm)	33.43	-	34.33	40.33	-	35.94	33.99
	% of Max	23.71	-	28.34	29.03	-	25.38	27.49
City Railway Station, Bengaluru - KSPCB	RMSE (ppm)	32.93	-	32.01	-	30.79	-	34.02
	% of Max	26.64	-	22.56	-	24.16	-	23.48
Hebbal, Bengaluru - KSPCB	RMSE (ppm)	35.54	34.90	33.51	34.06	38.51	34.93	33.37
	% of Max	29.96	26.44	26.96	27.48	29.01	25.48	26.98
Hombegowda Nagar, Bengaluru - KSPCB	RMSE (ppm)	39.88	35.28	35.22	42.52	36.76	35.25	36.32
	% of Max	29.05	27.02	28.22	30.85	26.29	24.16	26.40
Jayanagar 5th Block, Bengaluru - KSPCB	RMSE (ppm)	37.76	35.96	36.00	41.30	34.48	35.40	32.83
	% of Max	29.27	29.36%	25.66	27.90	26.32	28.86	23.60
Peenya, Bengaluru - CPCB	RMSE (ppm)	38.50	37.70	41.27	-	-	38.86	44.27
	% of Max	28.91	24.04	28.45	-	-	26.03	30.64
Sanegurava Halli, Bengaluru - KSPCB	RMSE (ppm)	42.44	-	35.62	-	32.12	-	38.14
	% of Max	34.84	-	28.49	-	27.77	-	30.50
Silk Board, Bengaluru - KSPCB	RMSE (ppm)	44.26	49.39	40.10	43.14	43.29	42.16	43.38
	% of Max	33.73	33.90	27.34	30.13	29.96	34.03	28.29

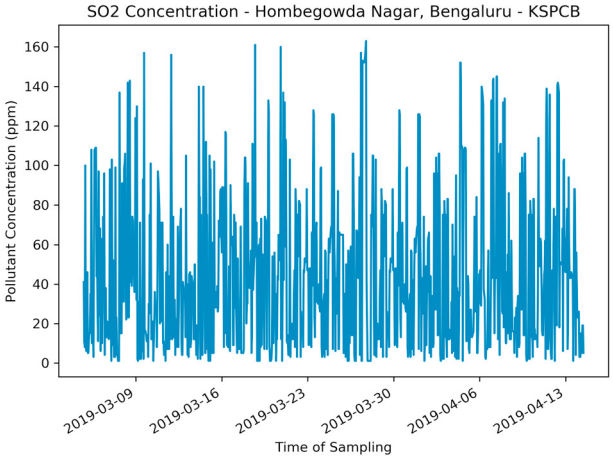

Fig. 4. SO_2 Data From Hombegowda Nagar, Bengaluru - KSPCB

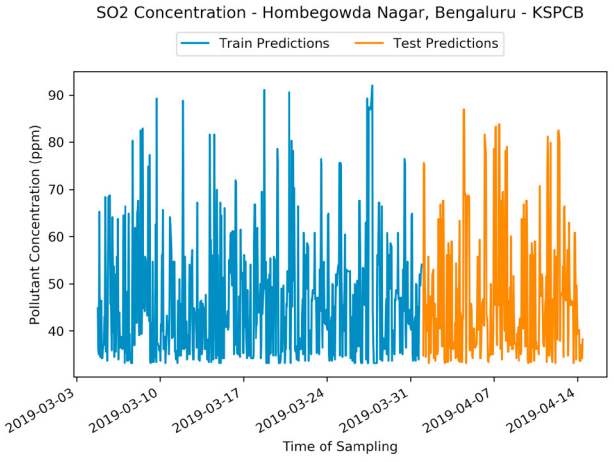

Fig. 5. SO_2 Predictions From Hombegowda Nagar, Bengaluru - KSPCB

5. Conclusions

From the above results it is observed that the forecasting model has an RMS error between 30-40 ppm in Bengaluru whereas it does significantly better in Amaravati where it has an RMS error between 0-5 ppm. From the graphs, it is also observed that data in Bengaluru being a huge metropolitan city has a very high temporal variance compared to data in Amaravati which is a small town. A high temporal variance in the data has an adverse effect on the performance of the forecasting model. From this, we can conclude that there does not exist a ”one fit for all” model that works in every city under any condition. To deal with the complex data such as in Bengaluru, we need to invest in more complex models that can reduce the error and improve the forecasting.

The model does not directly take factors such as wind, temperature, humidity and weather conditions that affect pollutant concentration into account. Taking these factors into account directly would improve the performance of the model.

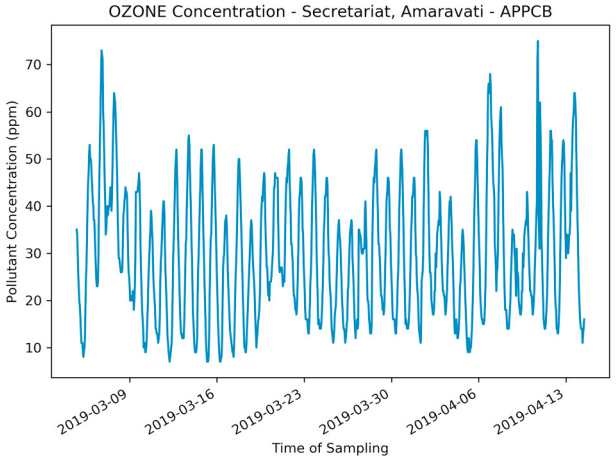

Fig. 6. O_3 Data From Secretariat, Amaravati - APPCB

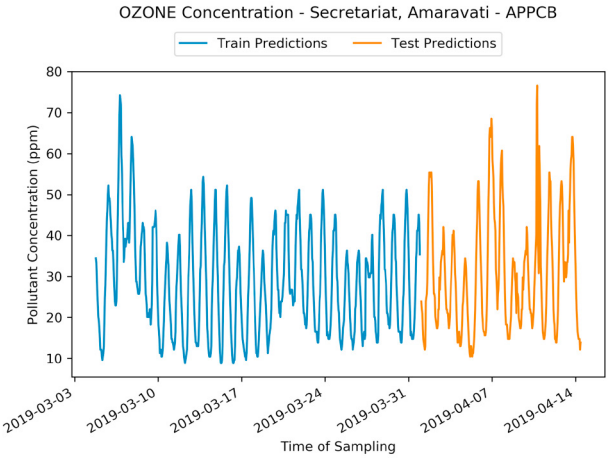

Fig. 7. O_3 Predictions From Secretariat, Amaravati - APPCB

The sensor nodes provide real time data of the pollutants, which can be used to bridge the gap between the hourly data that is given by the various Pollution Control Boards. These values cannot be taken with absolute confidence, however they can provide a good estimate with an accuracy of around $\pm 15 - 20\%$, depending on the quality of sensors used. Since cities typically have only a handful of government monitoring stations, these sensor nodes are a low cost, low power and easy to use alternative that can be installed in thousands of locations around the city and can be a good source of data for analysis, thus opening up many opportunities for real time data analysis of the acquired data.

Furthermore, the sensor nodes can be customised to provide some extra data about the concentrations of pollutants such as flammable gases and smoke, that are not provided by the government's monitoring stations.

References

[1] Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C., and Baklanov, A. *Real-time air quality forecasting, part I: History, techniques, and current status*, Atmos. Environ., 60, 632–655, 2012.

[2] G. Petnehazi, *"Recurrent neural networks for time series forecasting,"* arXiv preprint arXiv:1901.00069, 2019.

[3] *Real time Air Quality Index* data.gov.in/catalog/real-time-air-quality-index

[4] S. R. Jino Ramson and D. Jackuline Moni. *"Applications of wireless sensor networks — A survey"*. International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), 2017.

[5] Adenugba, et al. "Smart irrigation system for environmental sustainability in Africa: An Internet of Everything (IoE) approach". Mathematical Biosciences and Engineering, 16(5), 5490-5503. doi:10.3934/mbe.2019273

[6] Wei et al. 2019 "Multi-sink distributed power control algorithm for Cyber-physical-systems in coal mine tunnels". Computer Networks, 161, 210-219. doi:10.1016/j.comnet.2019.04.017

[7] Kadri, A.; Yaacoub, E.; Mushtaha, M.; Abu-Dayya, A. *"Wireless sensor network for real-time air pollution monitoring."*. In Proceedings of the 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Sharjah, UAE, 12–14 February 2013; pp. 1–5.

[8] Kavi K.Khedo, Rajiv Perseedoss, Avinash Mungur. *"A wireless sensor network Air pollution monitoring system"*. International Journal of wireless and Mobile Networks, Vol 2, 31-45, May 2010.

[9] Vendula Hejlova and Vit Vozenilek. *"Wireless Sensor Network components for Air Pollution in urban environment: Criteria and analysis for their selection"*. Wireless Sensor Networks, 229-240, 2103.

[10] Zhu, D., Cai, C., Yang, T., & Zhou, X. *"A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization."*. Big Data and Cognitive Computing, 2(1), 5, 2018.

[11] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber. *"LSTM: A search space odyssey"*. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222-2232, 2017.

[12] S. Hochreiter and J. Schmidhuber, *"Long short-term memory,"* Neural computation, vol. 9, pp. 1735–80, 12 1997.