

**ASIAN COLLEGE OF HIGHER STUDIES (ACHS)**

**Tribhuvan University**

**Institute of Science and Technology**



**A PROJECT REPORT**

**ON**

**AIR QUALITY PREDICTION**

**Submitted to**

**Institute of Science and Information Technology**

**Tribhuvan University**

**In Partial Fulfillment of the Requirement for the  
Bachelor Degree in Computer Science and Information Technology**

**Basanta Khadka(11031/073)**

**Nabin G.C(11049/073)**

**Umesh Sapkota(11070/073)**

**Utsav Dhungana(11071/073)**

## ACKNOWLEDGEMENT

We would like to thank our supervisor **Mr. Janak Kumar Lal** for his feedback and guidance during the course of this project. His contribution in simulating suggestions and encouragement helped us to coordinate our project.

Also, we would like to thank Program Coordinator for CSIT, Asian College of Higher Studies respected **Mr. Brihat Singh Boswa** for his inputs during the course of this project. He also kept us up-to-date with relevant notices and deadlines, which helped us stay on the track to complete this project.

At last, we would also like to thank all the teachers and faculty members who encouraged us and guided us time to time during the project.

.....

Mr. Janak Kumar Lal

Supervisor

## ABSTRACT

Air pollution and its prevention are constant scientific challenges during the last decades. However, they still remain huge global problems. Affecting human's respiratory and cardiovascular systems, they are the cause for increased mortality and increased risk for diseases for the population. Many efforts from both local and state governments are done in order to understand and predict air quality index aiming to improve public health. This project is one scientific contribution towards this challenge. In present days, Kathmandu, the capital city of Nepal, has a population of around 3.5 million. Rapid and haphazard urbanization in the last few decades have resulted in the degradation of the environment, predominantly in terms of air pollution. Thus, this project mainly focuses on the air condition and air quality situation of the Kathmandu valley. The purpose of this project is to develop a web-based application integrating some functionalities of air quality and weather forecasting that helps people to know about the current air quality condition of Kathmandu and its hourly air quality state. Air Quality Prediction System is the web-based platform that provides current, hourly and daily air quality prediction with which anyone can know about the air quality state and weather conditions of Kathmandu. The system also includes different Air Quality Index Comparisons. The system compares different air pollutants like ozone and PM2.5 (particulate matter) and forecast the Air Quality Index. After comparing different machine learning algorithms like Multi-Linear Regression, Random Forest, Neural Network and LSTM (Long Short-Term memory) for efficiency, the system uses Neural Network which predicts hourly, daily and current air quality index.

**Signature of Supervisor**

.....  
Mr. Janak Kumar Lal  
Asian College of Higher Studies

**Signature of HOD/ Coordinator**

.....  
Mr. Brihat Singh Boswa  
Asian College of Higher Studies

**Signature of External Examiner**

.....

# TABLE OF CONTENTS

|   |     |
|---|-----|
| ACKNOWLEDGEMENT   | i   |
| ABSTRACT  | ii  |
| TABLE OF CONTENTS   | iii |
| LIST OF TABLES  | v   |
| LIST OF FIGURES   | vi  |
| LIST OF ACRONYMS AND ABBREVIATIONS                        | vii |
| CHAPTER 1: INTRODUCTION                                   | 1   |
| 1.1 Introduction  | 1   |
| 1.2 Statement of Problem                                  | 1   |
| 1.3 Objectives  | 1   |
| 1.4 Scope and Limitations                                 | 2   |
| 1.4.1 Scope   | 2   |
| 1.4.2 Limitations   | 2   |
| 1.5. Methods  | 3   |
| 1.5.1 Data Collection                                     | 4   |
| Meteorological Data                                       | 4   |
| Air Pollutant Data  | 4   |
| 1.5.2 Feature Selection                                   | 5   |
| 1.5.3 Feature Extraction                                  | 6   |
| 1.6 Algorithms  | 7   |
| 1.6.1 Multiple Linear Regression                          | 7   |
| 1.6.2 Random Forest Regression                            | 7   |
| 1.6.3 Long Short-Term Memory (LSTM)                       | 8   |
| CHAPTER 2: STUDY OF EXISTING SYSTEM AND LITERATURE REVIEW | 11  |
| 2.1 Study of Existing Systems                             | 11  |
| 2.2 Literature Review                                     | 11  |
| CHAPTER 3: REQUIREMENT ANALYSIS                           | 14  |
| 3.1 System Requirement Analysis                           | 14  |
| 3.1.1 Functional Requirements                             | 14  |
| 3.1.2 Use Case Diagram                                    | 14  |
| 3.1.3 Non-functional Requirement                          | 16  |
| 3.1.4 System Requirement                                  | 16  |
| 3.2 Feasibility Study                                     | 16  |

|  |    |
|--|----|
| 3.2.1 Economic Feasibility                   | 17 |
| 3.2.2 Technical Feasibility                  | 17 |
| 3.2.3 Operational Feasibility                | 18 |
| 3.2.4 Schedule Feasibility                   | 18 |
| CHAPTER 4: SYSTEM DESIGN                     | 19 |
| 4.1 Sequence Diagram                         | 19 |
| 4.2 E-R Diagram                              | 20 |
| CHAPTER 5: SYSTEM IMPLEMENTATION AND TESTING | 21 |
| 5.1 Implementation Methodology               | 21 |
| 5.1.1 Multilinear Regression                 | 21 |
| 5.1.2 Random Forest                          | 22 |
| 5.1.3 LSTM (Long Short-Term Memory)          | 23 |
| 5.2 Analysis                                 | 27 |
| 5.2.1 Multi linear Regression                | 27 |
| 5.2.2 Random Forest Regression               | 27 |
| 5.2.3 LTSM Analysis                          | 29 |
| Trend - Seasonality-Residual Decomposition:  | 29 |
| Ad fuller test:                              | 29 |
| 5.2.4 Comparative analysis:                  | 31 |
| Post Deployment Results:                     | 32 |
| 5.3 Tools Used                               | 34 |
| 5.3.1 Analysis and Design Tools              | 34 |
| 5.3.2 Implementation Tools                   | 34 |
| 5.4 Testing                                  | 36 |
| 5.4.1 Unit Testing:                          | 36 |
| 5.4.2 Integration Testing:                   | 36 |
| CHAPTER 6: CONCLUSION AND FUTURE WORKS       | 38 |
| 6.1 Conclusion                               | 38 |
| 6.2 Future Works                             | 38 |
| REFERENCES                                   | 39 |
| APPENDIX                                     | 40 |

## LIST OF TABLES

|  |    |
|--|----|
| Table 5.1: MLR Prediction Analysis .....                                 | 27 |
| Table 5.2: RFR Prediction Analysis .....                                 | 27 |
| Table 5.3: LSTM Prediction Analysis .....                                | 30 |
| Table 5.4: Comparison of results of all algorithms .....                 | 31 |
| Table 5.5: Comparison of Real AQI and Predicted AQI after Deploying..... | 32 |
| Table 5.6: Test Case for different Units .....                           | 36 |
| Table 5.7: Test Case for Prediction Module.....                          | 37 |
| Table 5.8: Test Case for Past Data Module .....                          | 37 |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1.1 Workflow of AQI Prediction .....   | 3  |
| Figure 1.2 Dataset after feature selection and extraction .....                             | 6  |
| Figure 1.3 Schematic Representation of Random Forest Regression.....                        | 8  |
| Figure 1.4 LSTM-cell. It tracks and updates a cell state $c^{(t)}$ at every time-step ..... | 9  |
| Figure 1.5 Summary of LSTM cell equations .....   | 10 |
| Figure 3.1 Use-Case Diagram .....   | 15 |
| Figure 4.1 Sequence Diagram .....   | 19 |
| Figure 4.2 E-R Diagram .....  | 20 |
| Figure 5.1 LSTM Implementation Pipeline.....  | 23 |
| Figure 5.2 Real vs Predicted Values Train set using RFR .....                               | 28 |
| Figure 5.3 Real vs Predicted values Test set using RFR .....                                | 28 |
| Figure 5.4 Decomposition of PM2.5 into trend, seasonality and residuals .....               | 29 |
| Figure 5.5 Training vs Validation Loss .....  | 30 |
| Figure 5.6 Real vs Predicted values using LSTM .....  | 31 |

## LIST OF ACRYONMS AND ABBREVIATIONS

AQI: Air Quality Index

CART: Classification and Regression Trees

DNN: Deep Forward Neural Network

EDA: Exploratory Data Analysis

EPI: Environmental Performance Index

LSTM: Long Short-Term Memory

MAE: Mean Absolute Error

MLR: Multi Linear Regression

MART: Multiple Additive Regression Trees

MVT: Model View Template

PM: Particulate Matter

RFR: Random Forest Regression

RMSE: Root Mean Square Error

RNN: Recurrence Neural Network

SDLC: Software Development Life Cycle



## CHAPTER 1: INTRODUCTION

### 1.1 Introduction

Kathmandu Valley, well known as the city of temples, has now transformed itself into a city of pollution. The city of temples is now clad in dust and smoke. The pristine blue hills and the crisp blue sky that covered the valley just about two decades ago now appear gray and hazy due to the stagnant smog that hovers over them. The valley is surrounded by high mountains ranging from 2000 to 2800 meters from sea level. Due to this, the valley has a unique bowl-shaped topographic structure which restricts the movement of wind thereby retaining the pollutants in the air. This makes the valley particularly vulnerable to air pollution. Keeping track of this rapidly increasing pollution has been practiced in Nepal since 2009 and now we have nine monitoring stations currently in operation for measuring air quality in Kathmandu Valley. In our project we use a machine learning approach to predict ambient air quality with the help of meteorological and air quality data acquired from these stations.

### 1.2 Statement of Problem

Accompanying the rapid urbanization, many cities in developing countries including Nepal have led to environmental problems such as air pollution, water pollution, noise pollution and many more. Air pollution has a direct impact on human's health. Global warming, acid rains, increase in the number of asthma patients are some of the long-term consequences of air pollution. The demand for predicting future air quality is becoming increasingly more important to government's policy-making and people's decision making. Precised air quality forecasting can reduce the effect of maximal pollution on the humans and biosphere as well. Hence, enhancing air quality forecasting is one of the prime targets for the society.

### 1.3 Objectives

- To predict air pollution levels in a city with the ground data set.
- To show the relationships of air pollutant level with meteorological and traffic factors.
- To assist policy making process by providing insights about level of air pollutants.

## 1.4 Scope and Limitations

### 1.4.1 Scope

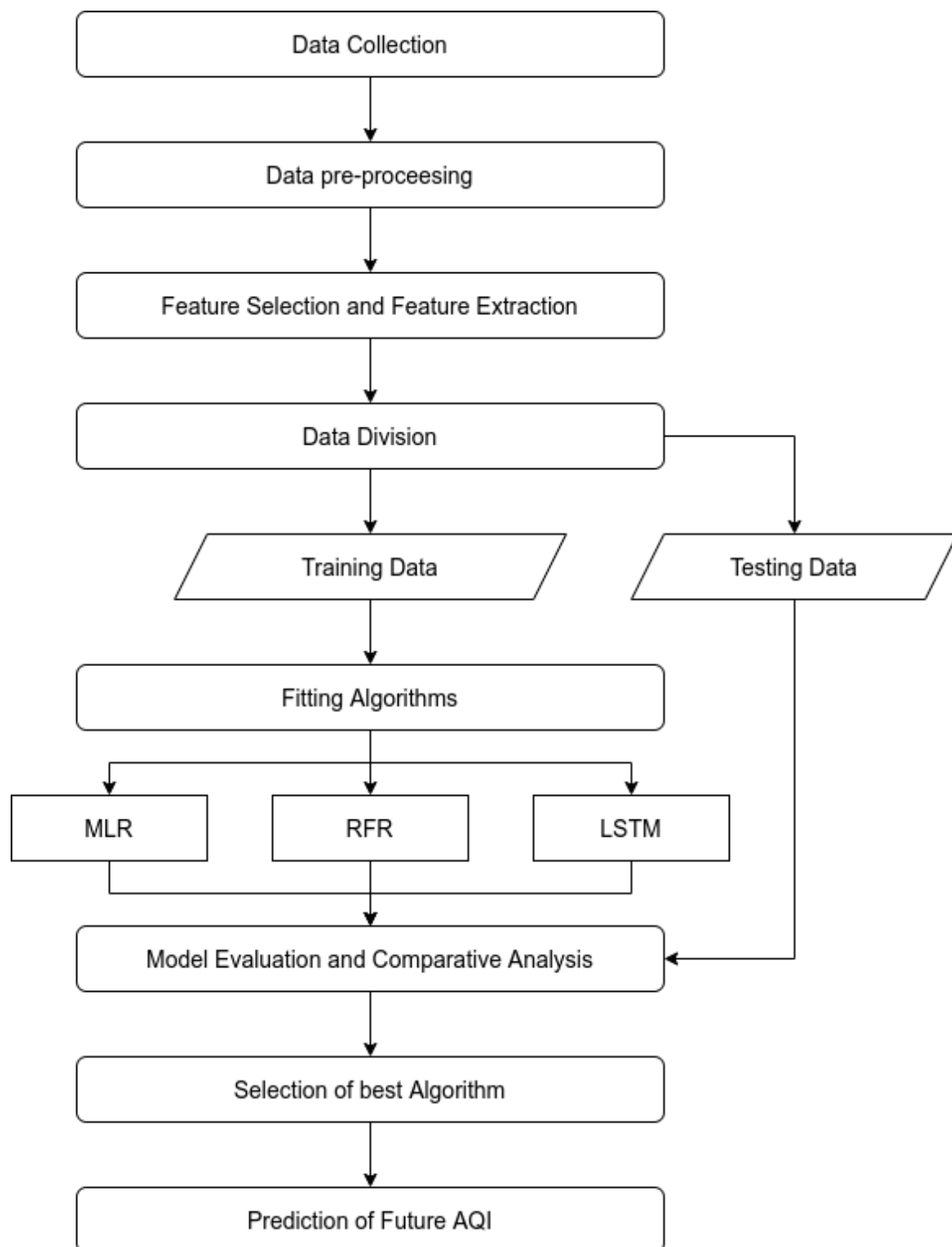
Air Quality Prediction System has the sole purpose of providing insights about daily air quality. It helps government and health organizations to be prepared for the hazardous effect of air pollution by spreading mass awareness. Currently our system is limited to forecast the air quality of Kathmandu city. It can further be refurbished to forecast the air quality of several cities according to the availability of data sets. Air quality prediction systems can be used by health organizations and also the health-conscious individuals can benefit from the air quality forecast. The policy making bodies of government can develop certain rules and regulations to control air pollution and also come up with pollution control technologies. It helps in establishing Air Quality goals according to pollution index in local or central government.

### 1.4.2 Limitations

- Currently the system is able to predict the ambient pollution level only in Kathmandu city due to data unavailability.
- Considering data availability, traffic factors and industrial parameters playing significant roles in air pollution could not be included.
- Data sets used are expected to achieve at least 80% accuracy which could be further increased with larger datasets.
- Time series data needs to be retrained every time we need to generate new predictions
- The uncertainty of the forecast is just as important as even more so, than the forecast itself
- If you have less data points we have to do away with train/test splits

### 1.5. Methods

The proposed methods will follow through the following phases:



*Figure 1.1 Workflow of AQI Prediction*

### 1.5.1 Data Collection

Two types of data will be used for the prediction of Air Quality Index in Kathmandu:

#### Meteorological Data

- Meteorology/Weather factors like Temperature, Wind Speed, Relative Humidity, Speed and Direction of wind, atmospheric pressure etc. affect the quality of air so the relationship/impact of weather on Air Quality is studied using a weather data-set.
- Meteorological data will be obtained from the Department of Hydrology and Meteorology, Government of Nepal by using their request feature of the website <http://www.dhm.gov.np/requestfordata/> and by making a certain payment.
- The Department of Hydrology and Meteorology was established in 1967 AD and is the primary source for our data. It is the most reliable and accurate source of data with 30+ years of historical weather data for Kathmandu.
- The repository contains historical data from 1967 but we will be using data only from 2017 to 2021 with hourly/daily intervals

The repository contains following parameters:

- a) Temperature (Maximum/ Minimum/ Mean)
- b) Relative humidity (Morning/ Evening)
- c) Rainfall
- d) Wind (Speed/ Direction)
- e) Evaporation
- f) Soil temperature (per depth)
- g) Grass minimum temperature
- h) Pressure
- i) Soil moisture etc.

#### Air Pollutant Data

- Air pollution factors like PM 10, PM 2.5, O3 etc. are the particles that adversely affect the quality of air hence the relationship is studied using air pollutant data-set.
- The air pollutant data is extracted from US embassy of Nepal which has installed two air quality monitoring stations US embassy building and Phora Durbar
- Data is downloaded from following websites in CSV format

[https://www.airnow.gov/international/us-embassies-and-consulates/#Nepal\\$Embassy\\_Kathmandu](https://www.airnow.gov/international/us-embassies-and-consulates/#Nepal$Embassy_Kathmandu)

[https://www.airnow.gov/international/us-embassies-and-consulates/#Nepal\\$Phora\\_Durbar\\_Kathmandu](https://www.airnow.gov/international/us-embassies-and-consulates/#Nepal$Phora_Durbar_Kathmandu)

- The repository contain data from 2017-2020 from which we will be using 3 years' data from January 2017 to July 2020
- The data-sets contains data about following parameters in hourly, daily and monthly intervals:
  - a. PM 2.5
  - b. PM 10
  - c. Ozone(O3)
  - d. Air Quality Index (AQI)

- Real time data at daily interval is downloaded from AQICN website to make future predictions at regular intervals

<https://aqicn.org/city/nepal/kathmandu/us-embassy/>

### 1.5.2 Feature Selection

Feature Selection is the process of choosing a subset of the relevant features for use in model construction. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on. The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors and providing a better understanding of the underlying process that generated the data.

Following are the original sets of features that we acquired after data collection:

1. NowCast Conc (Ozone)
2. Raw Conc (Ozone)
3. Processed Ozone value
4. Nowcast Conc (PM2.5)
5. Raw Conc (PM2.5)
6. Processed PM2.5
7. Processed PM 10
8. Relative humidity at UTC hour 3 and 12
9. Minimum temperature
10. Maximum temperature
11. Total Rainfall daily

We performed correlation analysis on these set of features and based on the results we selected following features on the basis of relevancy

1. Processed Ozone
  2. Processed PM2.5
  3. Processed PM10
  4. Total Rainfall
  5. Relative Humidity
  6. Minimum Temperature
  7. Maximum Temperature
- Raw Conc. and Nowcast Conc. was dropped to remove redundancy since they were processed to get the ozone and pm2.5 values.
  - Only PM2.5 values were selected for univariate time analysis in LSTM

### 1.5.3 Feature Extraction

Feature Extraction is done on an initial set of obtained data and builds derived values(features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps thus creating a better model for prediction. Feature extraction builds valuable information from raw data- the features -by reformatting, combining transforming primary features into new ones ... until it yields a new set of data that can be consumed by the Machine Learning models to achieve their goals. It was done in the following ways:

- Reformatting all the features to daily intervals by aggregating over hourly intervals
- PM 2.5 was converted into a dataset of  $t$ ,  $t-1$ ,  $t-2$ ,  $t-3$ , .....,  $t-n$

Where,  $t$  = label or current value

$n$  = no of time steps to look back

It was done to do univariate time series prediction using LSTM

- Rolling window method was used for preliminary analysis and to remove noise in the data
- Day, month, year was combined into a single date time index

|            | Rainfall | Tmax | Tmin | relative_humidity | pm10 | Ozone     | Pm2.5 |
|------------|----------|------|------|-------------------|------|-----------|-------|
| date       |          |      |      |                   |      |           |       |
| 2017-03-23 | 0.0      | 30.5 | 9.0  | 83.50             | 69.0 | 46.041667 | 145.0 |
| 2017-03-24 | 0.0      | 31.0 | 9.5  | 85.95             | 69.0 | 51.250000 | 149.0 |
| 2017-03-25 | 0.0      | 31.5 | 9.5  | 91.45             | 69.0 | 50.083333 | 104.0 |
| 2017-03-26 | 16.0     | 29.2 | 10.2 | 88.80             | 69.0 | 46.750000 | 159.0 |
| 2017-03-27 | 2.0      | 28.5 | 11.2 | 86.75             | 69.0 | 51.625000 | 158.0 |
| ...        | ...      | ...  | ...  | ...               | ...  | ...       | ...   |
| 2020-06-26 | 19.1     | 28.4 | 20.0 | 88.60             | 22.0 | 26.000000 | 22.0  |
| 2020-06-27 | 1.2      | 30.0 | 20.5 | 82.65             | 25.0 | 31.375000 | 35.0  |
| 2020-06-28 | 0.0      | 27.2 | 20.7 | 88.65             | 22.0 | 30.083333 | 42.0  |
| 2020-06-29 | 3.2      | 29.2 | 20.5 | 83.35             | 37.0 | 29.250000 | 37.0  |
| 2020-06-30 | 6.0      | 30.7 | 20.6 | 75.45             | 35.0 | 28.625000 | 63.0  |

1196 rows × 7 columns

*Figure 1.2 Dataset after feature selection and extraction*

## 1.6 Algorithms

### 1.6.1 Multiple Linear Regression

Multiple Linear Regression is the simplest linear regression model. It is based on a large number of experiments and observations to find the relationship between dependent and independent variables. Model parameters are given to determine the relationship between air quality index and features. The multiple linear regression was used in [2], [3] and [8] and average error was 5-12%. Although the error by using this algorithm is higher than other advanced algorithms, it provides a great starting point to conceptualize the relationship and implement other algorithms to improve accuracy. The multiple linear regression model is given by Eq. (1).

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n$$

Where  $y$  denotes the dependent variable,  $x_1, x_2, x_3, \dots, x_n$  are independent variables,  $w_1, w_2, w_3, \dots, w_n$  are model parameters learned from the training data.

### 1.6.2 Random Forest Regression

Random forests (RF) are an ensemble learning method for classification, regression and other tasks. A RF operates by constructing multiple decision trees at different training times, and outputting the class representing the mode of classes (classification), or mean prediction (regression) on individual trees.

The RF algorithm incorporates growing classification and regression trees (CARTs). Each CART is built using random vectors. For the RF based classifier model, the main parameters were the number of decision trees as well as the number of features ( $N_f$ ) in the random subset at each node in three growing trees. During model training the number of trees were determined at first. Larger number of trees is better but requires longer time to compute.

Lower the number of features higher the reduction in variance, but increase in bias.

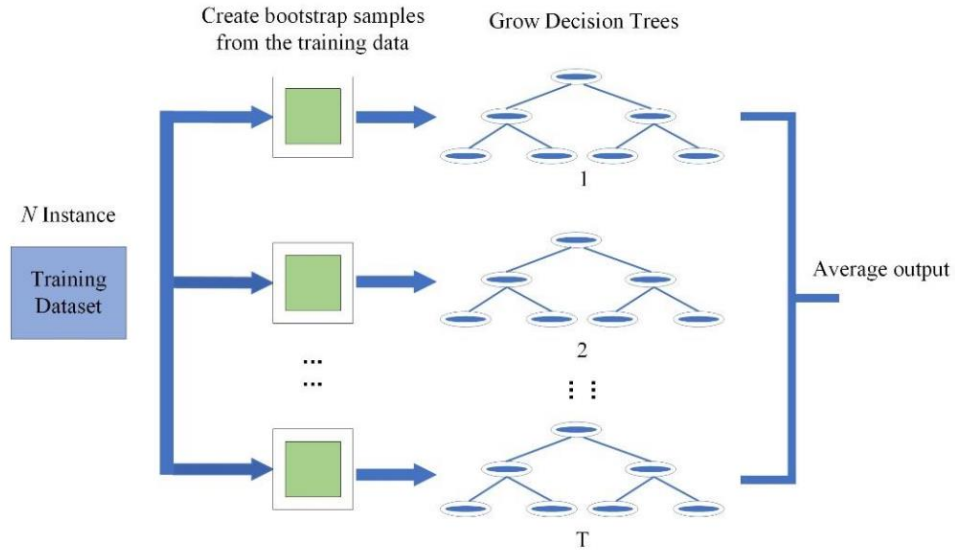
RF can be applied to classification and regression problems, depending on the requirement i.e. classification or regression tree.

Assuming that the model includes  $T$  regression trees for regression prediction the final output of the model is given by the mean of  $T$  regression trees as given below:

$$H(x) = \frac{1}{T} \sum_{i=1}^T h_i(x),$$

Where  $T$  is the number of regression trees, and  $h_i(x)$  is the output of  $i^{\text{th}}$  regression tree on sample  $x$ .

The regression model as shown by figure below:



*Figure 1.3 Schematic Representation of Random Forest Regression*

Random forest regression (RFR) is implemented in [2], [3], [6]. The MAE and RMSE in [2] were 4.4 and 6.4 respectively.

MAE, RMSE and R in [3] was 7.3, 9.5, 0 and 991 respectively and on [6] RMSE and  $R^2$  was 7.666 and 0.976 respectively.

The error rate was about 4-10% on average. It also outperformed Back Propagation Neural Nets (BPNN) and Multilayer Perceptron's (MLP) with proper optimizations hence, this algorithm was selected for AQI prediction.

### 1.6.3 Long Short-Term Memory (LSTM)

The LSTM is a type of Recurrent Neural Network (RNN) which can exhibit temporal dynamic behavior for a time sequence. LSTM takes not only the current data as the input but also considers what it got previously. So, the input of the LSTM model at time  $t$  is the model output at time  $t-1$  along with the new output at time  $t$ . Unlike a feed-forward neural network it can apply the internal state (memory unit) to judge whether the information is useful or not. In addition to that unlike traditional RNN is capable of learning long term dependence and is not affected by vanishing gradient problems. Therefore, the LSTM is utilized for this project.



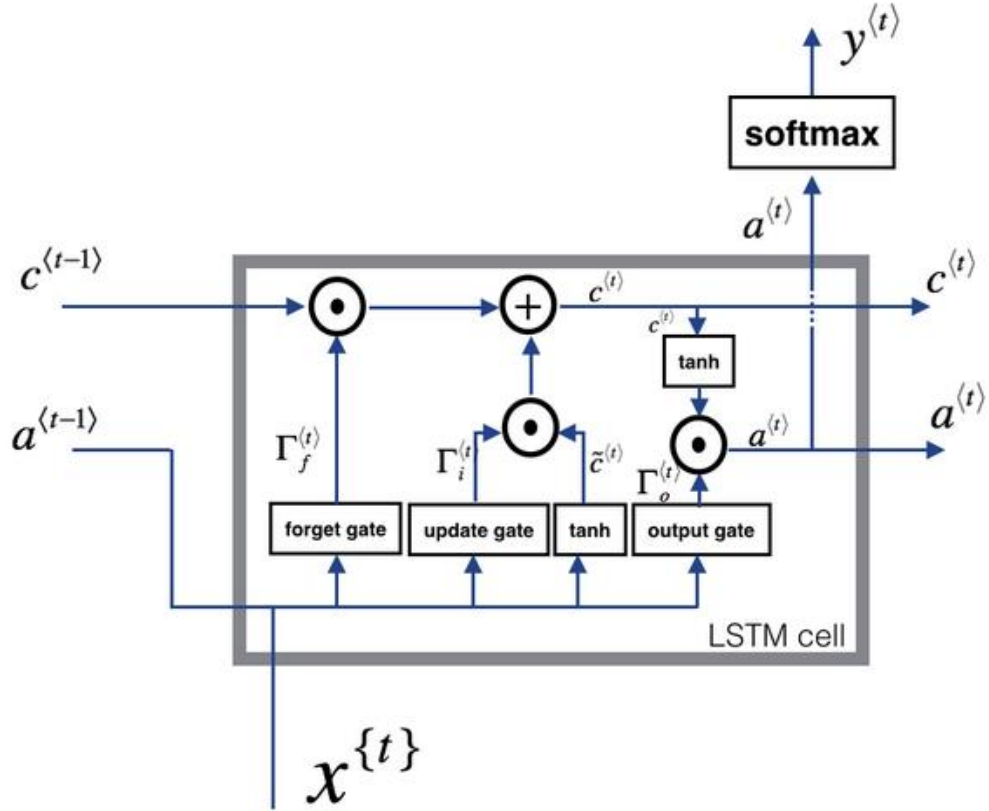


Figure 1.4 LSTM-cell. It tracks and updates a cell state  $c^{(t)}$  at every time-step, which can differ from  $a^{(t)}$

### Forget Gate

Forget gate uses the previous output  $a^{(t-1)}$  current input  $x_{(t)}$  to give output  $\Gamma_f$ . Then, the function sends the output to the current cell called  $c_{(t)}$ .

$$\Gamma_f = \sigma(W_f[a^{(t-1)}, x^{(t)}] + b_f) \quad (1)$$

Here  $W_f$  and  $b_f$  are the weights and bias of forget state, and  $\sigma$  represents the sigmoid function. The value of  $\Gamma_f$  lies between 0 and 1. This forget gate vector will be multiplied element-wise by the previous cell state  $c^{(t-1)}$ . So if one of the values of  $\Gamma_f$  is 0 (or close to 0) then it means that the LSTM should remove that piece of information (e.g. the singular subject) in the corresponding component of  $c^{(t-1)}$ . If one of the values is 1, then it will keep the information.

### Update Gate

Update gate and tanh function coordinate control how much new information is added to the cell state. The formula for update gate:

$$\Gamma_u = \sigma(W_u[a^{(t-1)}, x^{(t)}] + b_u) \quad (2)$$

$W_u$  and  $b_u$  are the weights and bias of update state.  $\Gamma_u$  is a value between 0 and 1 which will be multiplied element wise with  $\tilde{c}^{(t)}$  in order to compute  $c^{(t)}$

To update the new subject, we need to create a new vector of numbers that we can add to our previous cell state. The equation is:

$$\tilde{c}^{(t)} = \tanh (W_c [a^{(t-1)}, x^{(t)}] + b_c) \quad (3)$$

Finally, the new cell state is:

$$c^{(t)} = \Gamma_f * c^{(t-1)} + \Gamma_u * \tilde{c}^{(t)} \quad (4)$$

Where  $W_c$  and  $b_c$  are the weights and bias of cell state, '\*' means element-wise multiplication and '+' represents element-wise addition. Equation 2 and 3 calculate the information to be updated and equation 4 realizes the cell state update.

### Output gate

Output gate (OG) controls the current information in the cell state to flow into the outputs. It outputs the value that we need to output based on the cell state, but also the filtered version. First of all, we run a sigmoid function to determine which part of the cell state will be output. Then we manipulate the cell state through tanh (to get a value between -1 and 1) and multiply it by the output of the sigmoid gate.

$$\Gamma_o = \sigma (W_o [a^{(t-1)}, x^{(t)}] + b_o) \quad (5)$$

Where  $W_o$  and  $b_o$  are weights and bias of the output gate

$$a^{(t)} = \Gamma_o * \tanh (c^{(t)}) \quad (6)$$

To summarize all the equations for 't' cells is:

$$\begin{aligned} \Gamma_f^{(t)} &= \sigma(W_f[a^{(t-1)}, x^{(t)}] + b_f) \\ \Gamma_u^{(t)} &= \sigma(W_u[a^{(t-1)}, x^{(t)}] + b_u) \\ \tilde{c}^{(t)} &= \tanh(W_c[a^{(t-1)}, x^{(t)}] + b_c) \\ c^{(t)} &= \Gamma_f^{(t)} \circ c^{(t-1)} + \Gamma_u^{(t)} \circ \tilde{c}^{(t)} \\ \Gamma_o^{(t)} &= \sigma(W_o[a^{(t-1)}, x^{(t)}] + b_o) \\ a^{(t)} &= \Gamma_o^{(t)} \circ \tanh(c^{(t)}) \end{aligned}$$

Figure 1.5 Summary of LSTM cell equations

## CHAPTER 2: STUDY OF EXISTING SYSTEM AND LITERATURE REVIEW

### 2.1 Study of Existing Systems

Information and communication technologies are rapidly developing. Broadly implemented web-based applications such as the Air Quality Prediction System has potential to overcome pollution related issues and make people aware.

There are different web-based and mobile applications already available such as [airnow.gov](http://airnow.gov), [aqcin.org](http://aqcin.org), [accuweather.com](http://accuweather.com), and [iqair.com](http://iqair.com) that helps to predict and know about air quality conditions as well as weather forecasting.

Despite being an emerging and well-known project, most of these applications are globally bounded and are not limited to a particular area. Also, accuracy in prediction of AQI index seems slightly distorted. Majority of these applications lack past AQI information and pollution parameters' information so that people know exactly what has led to such AQI variation.

Hence, with the intent of providing a particular area bound application and to give people information about pollution parameters and AQI variation, the concept of Air Quality Prediction of Kathmandu comes into existence.

### 2.2 Literature Review

Air pollution is a complex mixture of thousands of components, majority of which include airborne Particulate Matter (PM) and gaseous pollutants like ozone (O<sub>3</sub>), nitrogen dioxide (NO<sub>2</sub>), volatile organic compounds (like benzene(C<sub>6</sub>H<sub>6</sub>)), carbon monoxide (CO), sulfur dioxide (SO<sub>2</sub>), etc. Air pollution is considered to be one of the biggest environmental threats. The air pollution problem in most cities is severe and has become the focus of the public and government. Entire cities had to be shut down due to high concentration of air pollution.

Till date, the majority of studies on impacts of air pollution on human health have been done in North America and Europe. Only few studies on this regard have been done in regions like Nepal. In 2016, the Environmental Performance Index (EPI) of Nepal's air quality ranked 177th out of 180 countries. The air pollution problem is highly geographical so each city has to be individually studied in order to know the pollution scenario. The valley is surrounded by high mountains ranging from 2000 to 2800 meters from sea level. Due to this, the valley has a unique bowl-shaped topographic structure which restricts the movement of wind thereby retaining the pollutants in the air. This makes the valley particularly vulnerable to air pollution.[\[1\]](#)

Most studies predict air quality index (AQI) using meteorological or air pollution factors thus prediction accuracy was low in the absence of complex mathematical models optimized for the terrain. To solve this problem both meteorological and air pollution factors were combined and used as teacher training data.[2]

Bo liu and Chao Shi in [3] makes the argument that most of the studies focus only on simple regression problems while the air quality is a multi-factor control problem hence various machine learning models were compared and the model with best performance was selected for AQI prediction.

Generally, the air pollution models simulate the concentrations of the pollutants in specific locations and at specific moments according to the real concentrations monitored by some stations over the city.[4]

The previous models for AQI prediction can be grouped into three categories (1) simple empirical approaches, (2) physically-based approaches, and (3) parametric and non-parametric statistical approaches. The simple approaches either use values of the present day to predict those of tomorrow or strictly depend on meteorological variables or forecasted pollutants. Physically based approaches on the other hand are more accurate than simple empirical approaches mainly due to their capability of modeling temporal and spatial patterns of meteorological and air pollutants. However, these processes are too complex to represent by physically-based models which results in biased forecasts. Parametric and non-parametric statistical approaches such as neural networks can outperform physically-based approaches in the accuracy of forecasts.[5]

The characteristics of real PM 2.5-time series exhibit non linearity and time varying complexity due to multivariate impacts. Therefore non-linear models are increasingly considered as alternatives to linear ones[6]. Yun Bai , Bo Zeng proposed in their paper [6] that ensemble based LSTM provides the best forecasting performance on the basis of mode decomposition, local modeling and ensemble learning providing the best degree of generalization.

The works of Mohit Bansal, Tanishq Verma and Anirudh Agrawal show that deep learning based models are promisingly better than conventional systems. It makes a strong argument that traditional approaches that use statistical and mathematical techniques for air quality prediction are inefficient, complex and provide limited accuracy. The temporal sequences of four meteorological parameters and pollutant level at hourly intervals gave

useable results and showed scalability with higher number of future time steps and more pollutant parameters [7]

Vidushi Chaudhary, Anand Deshbhratar also used deep learning based LSTM model to predict future air pollutant concentration by treating the problem as a time series based problem where current pollutant levels are dependent on previous pollutant, meteorological, traffic data festivals and holidays information.[8] The proposed system predicted the pollutant concentration for the next 12 hours with acceptable error range.

Comparison of multiple additive regression trees (MART), deep forward neural network (DNN) and hybrid model based on LSTM done in [9] and observed that by capturing temporal dependencies in different intervals, LSTM achieved best results by predicting 75% of the pollution levels over the next 58 hours.

The papers [6], [7], [8] and [9] make use of LSTM extensively to predict the AQI and PM 2.5 mostly at hourly intervals and show that they outperform traditional methods so, we will be implementing LSTM and making comparisons with other machine learning models for prediction of AQI in Kathmandu city.

Finally the major limitation to Air Quality Index Prediction is that the application of the results is highly local due to specific chemical and meteorological factors so it cannot be applied to other cities but we can take data from other cities and with minor adjustment can make predictions to those stations.[2, 5]

Accurate air quality forecasting has important theoretical and practical value for the public; without it, neither the government nor the public can effectively avoid the health damage caused by air pollution or improve the emergency response capability of heavy pollution days.

## CHAPTER 3: REQUIREMENT ANALYSIS

### 3.1 System Requirement Analysis

The system requirement analysis is done in order to acquire the details of the system and desired functionality of the system efficiently which includes both the hardware and software requirement for the satisfactory output and functionality of the application.

The requirements are divided into functional and non-functional requirements. In functional requirement, the activities and services that system must provide were included which may be in terms of input, output and processing. In the non-functional requirement the system's performance, accuracy and flexibility are included.

#### 3.1.1 Functional Requirements

Some functional requirement for our system includes:

1. Prediction Engine  
Users can predict AQI using the prediction engine.
2. Past Data  
User can explore past data and graphs
3. Download Data  
User can download past data in csv format
4. Signup and login  
User must be able to register and must be logged in to download the data

#### 3.1.2 Use Case Diagram

A use case diagram is a graphical representation of the interaction among the elements of a system. It captures the dynamic aspects of system functionalities. We are using the use case as a methodology for system analysis to identify, clarify, and organize our system requirements.

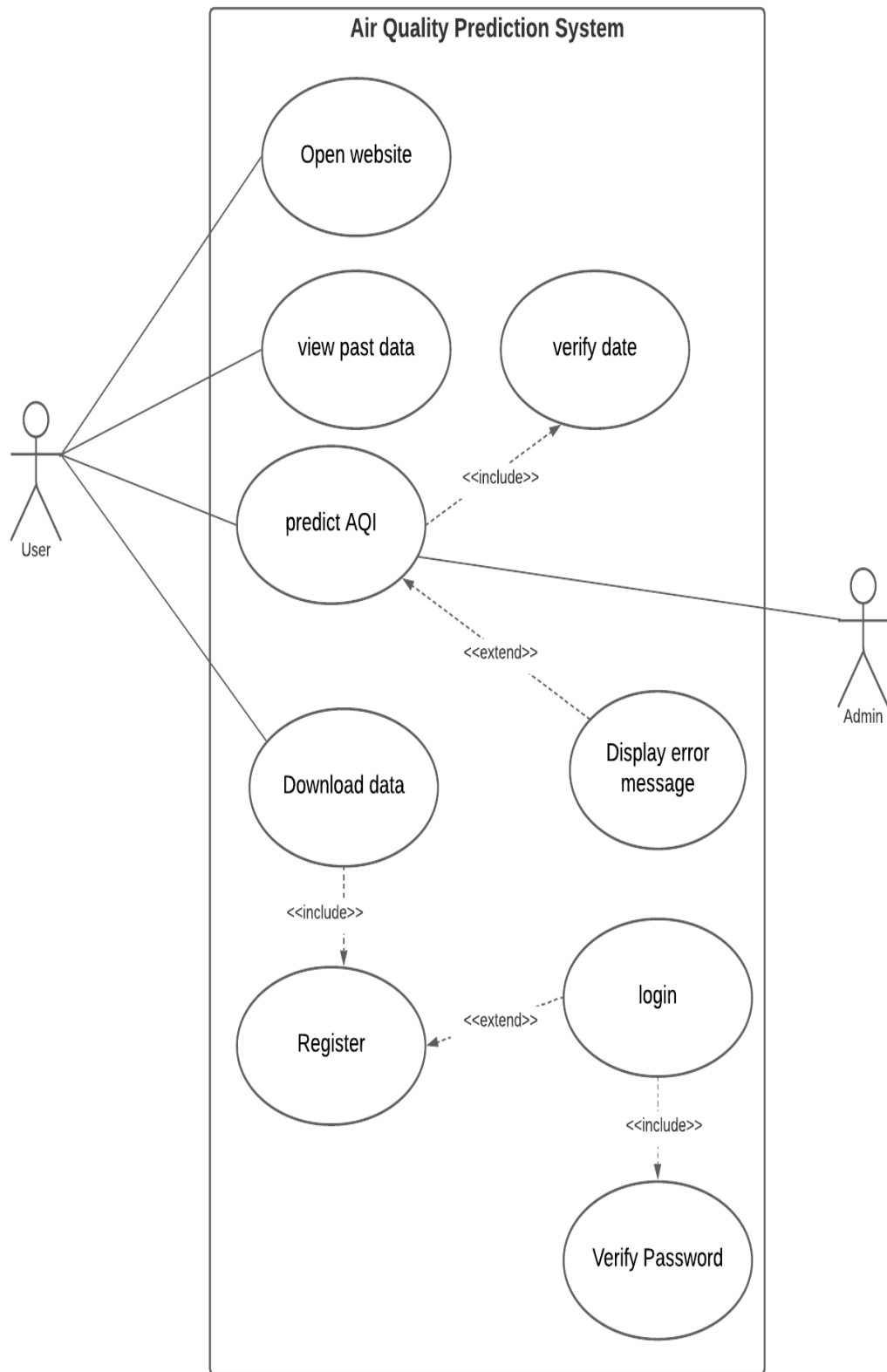


Figure 3.1 Use-Case Diagram

### 3.1.3 Non-functional Requirement

Non- functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the developing process and standards. Non-functional requirements often apply to the system as a whole.

1. Performance: We are training the model to correctly categorize contents with an accuracy of more than 90%.
2. Functionality: This software will deliver on the functional requirements mentioned in the document.
3. Availability: This system will be deployed using DJANGO database tools that ensures high availability of backend data and the system is web-based.
4. Learn ability: The software is very easy to use and reduces learning work.
5. Reliability: This software will work reliably for low space.

### 3.1.4 System Requirement

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. The following are some of the requirements for our Air Quality Prediction System.

1. Front End
  - Html
  - CSS
  - JavaScript.
2. Back End
  - DJANGO

Django is a Python-based free open-source web framework that follows the model-template- view (MVC) architectural pattern. We are using Django to create our web-based application and interaction of Regression algorithms with the application.

## 3.2 Feasibility Study

Feasibility study is the detailed process of working out whether the proposed system or proposed plan or piece of work is economically and technically applicable and justifiable. Sometimes the project is not even reasonable hence a detailed study on the viability of the project should be carried out before making any advancements towards technical development of the project. Feasibility Study is done to determine whether the



proposed system is viable considering the Technical, Operational and Economic factors. After going through feasibility study, we can have a clear-cut view of system's benefits and drawbacks. We will perform three basic feasibilities each of them is described below:

### 3.2.1 Economic Feasibility

It is the measure of cost-effectiveness of the project. It involves capital cost of starting the project, detailed operational cost and the legacy cost projection. The proposed system is economically feasible because of following reasons:

- Some minimal cost will be incurred to acquire meteorological data from Department of Meteorology, Nepal Government.
- Air pollutants data has been made freely available by US embassy, Nepal
- No additional cost to acquire new hardware or software since models will be trained on our own machines
- The resources required for the development of the project are nominal and so is the starting cost of the project.
- Our project is financially very sound and operable, with low maintenance costs, if any, which may emerge in the future.

### 3.2.2 Technical Feasibility

It is the measure of practicality of a specific technical solution and availability of a specific technical solution and availability of technical resources and expertise. It addresses the hardware and software considerations.

- No new hardware is needed and all the software that will be used are open-source free software
- All the machine learning libraries that will be used are open-source and well within our technical knowledge of implementation
- Python will be used as the programming language in prediction of AQI
- The results will be displayed in web interface using Django framework
- EDA and visualization will be used to explore and visualize data

### 3.2.3 Operational Feasibility

It is the measure of how well the system will work once deployed. It is concerned with how end users of the system feel about the system. The proposed system is operationally feasible due to following reasons:

- End users can easily get information about present and forecasted AQI and can take actions to protect their health
- An intuitive and simple interface to visualize historical and forecasted data

Local government and other environmental protection organizations can also implement this system to predict AQI and plan their future course of action

### 3.2.4 Schedule Feasibility

|                    | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
|--------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|
| Study and Analysis |        |        |        |        |        |        |        |        |        |         |         |         |
| Data Collection    |        |        |        |        |        |        |        |        |        |         |         |         |
| Implementation     |        |        |        |        |        |        |        |        |        |         |         |         |
| Testing            |        |        |        |        |        |        |        |        |        |         |         |         |
| Documentation      |        |        |        |        |        |        |        |        |        |         |         |         |
| Review             |        |        |        |        |        |        |        |        |        |         |         |         |
| Presentation       |        |        |        |        |        |        |        |        |        |         |         |         |

## CHAPTER 4: SYSTEM DESIGN

We have developed the architecture for the system that shows how different services in the system interact to provide collective functionalities.

### 4.1 Sequence Diagram

A Sequence diagram shows interaction between the external actors and system. It represents how objects operate with one another and in what order. It is the object interactions arranged in time sequence. The following sequence diagram depicts the flow of information in our air quality prediction system.

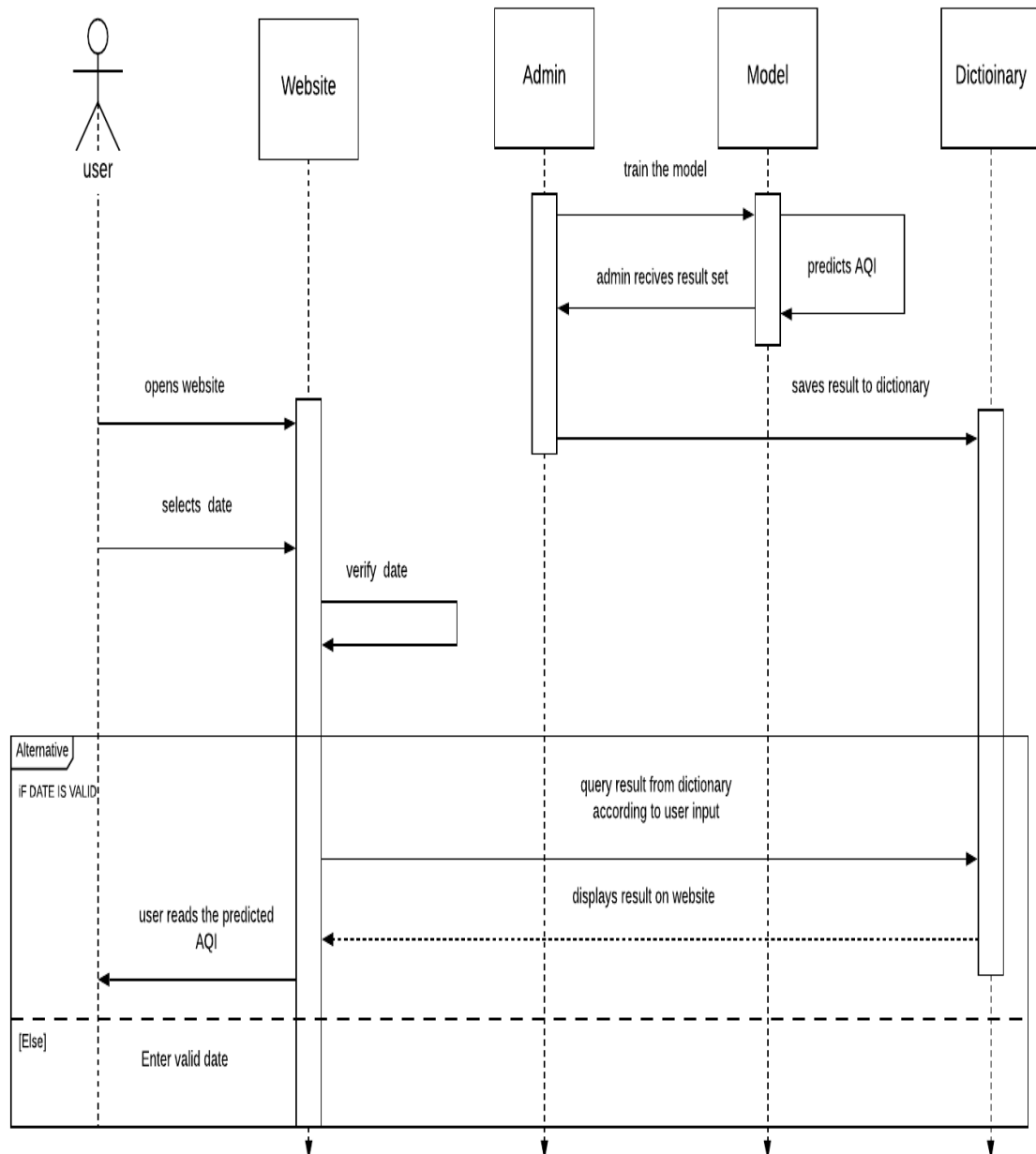


Figure 4.1 Sequence Diagram

## 4.2 E-R Diagram

Entity Relationship Diagram (ERD or ER Diagram or ER model) is a type of structural diagram used in database design. The following ERD contains different symbols and connectors that visualize the major entities within the Air Quality Prediction System scope, and the interrelationships among these entities.

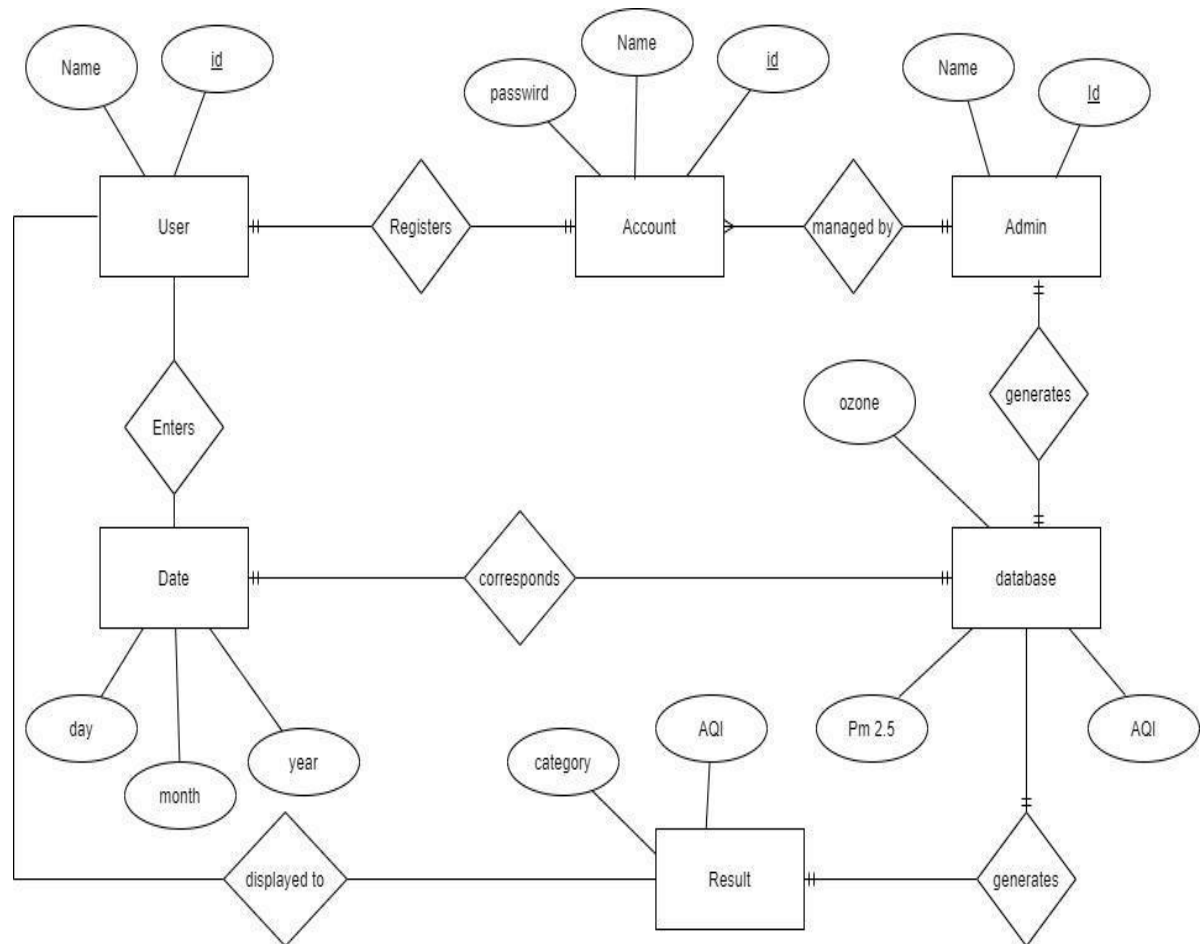


Figure 4.2 E-R Diagram

## CHAPTER 5: SYSTEM IMPLEMENTATION AND TESTING

### 5.1 Implementation Methodology

System implementation is the process of defining how the information system is built, ensuring that the information system is operational and meets the quality standard. During the research period various methodologies were used for product requirements and data collection. The primary data and information about the project were collected through web scraping and manually doing research on the relevant field. Different algorithms were gathered and analyzed for best performance in our system and the best algorithm was selected for deployment to our website. The website was developed using Django framework and SQLite database.

#### 5.1.1 Multilinear Regression

1. Start
2. Load the filtered dataset into dataframe  
`df = pd.read_csv('filtered_data.csv')`
3. Check the top values of dataframe.  
`df.head()`
4. Check columns of the dataframe and check for errors
  - a. Yes, review dataset and resolve the errors
  - b. No, go to step 5
2. Clean the dataframe 'df'
3. Split the dataset into features and labels  
`feature_set = df.iloc[:, :-1]`  
`label_set = df.iloc[:, -1]`
4. Normalize the features  
`feature_set = (feature_set - feature_set.mean()) / (feature_set.max() - feature_set.min())`
5. Visualize the dataset using a pairwise plot of independent and dependent variables.
6. Split the dataset into train and test data  
`X_train, X_test, y_train, y_test = train_test_split(feature_set, label_set, test_size=0.3, random_state = 42)`
7. Train the multilinear regression on training dataset  
`linear_regression_model(X_train, y_train, X_test, y_test, learning_rate = 0.05, epochs=200)`
8. Calculate the error metrics Mean Absolute Error
9. Check if error is satisfiable:
  - a. If yes, go to step 13
  - b. Else, revise model and retrain
10. Make predictions of unseen data
11. Serialize the model and save it  
`filename = 'finalized_model.sav'`  
`pickle.dump(nb, open(filename, 'wb'))`
12. End

### 5.1.2 Random Forest

1. Start
2. Load the filtered dataset into dataframe  
`df = pd.read_csv('filtered_data.csv')`
3. Check the head of dataframe  
`df.head()`
4. Preprocess the data frame
  - a. Sort values by date
  - b. Check null values in each column
5. Split the dataset into training and validation using `split_point`
  - a. `train_df = df.iloc[ : split_point]`
  - b. `val_df = df.iloc[split_point : ]`
6. Perform grid search on the training dataset
  - a. `grid_search = pd.DataFrame(grid_search)`
  - b. `grid_search.sort_values("r_squared_val", ascending = False).head()`
7. Create a tree using best `best_max_depth` and `best_min` samples returned from `grid_search`  
`tree = decision_tree_algorithm(train_df, ml_task="regression", max_depth=best_max_depth, min_samples=best_min_samples)`
8. Create a forest object by on `train_df`  
`forest = random_forest_algorithm(train_df, n_trees, n_bootstrap, n_features, dt_max_depth, dt_min_samples)`
9. Check your forest structure  
`print(forest)`
10. Make predictions on validation set using forest object  
`predictions = random_forest_predictions(va_df, forest)`
11. Check if error is acceptable
  - a. `mean_absolute_error(y_test, predictions)`
  - b. `np.sqrt(mean_squared_error(y_test, y_pred))`  
If acceptable go to step 12  
Else, revise model and go to step 5
12. Serialize the model and save it
  - a. `filename = 'finalized_model.sav'`
  - b. `pickle.dump(nb, open(filename, 'wb'))`
13. End

### 5.1.3 LSTM (Long Short-Term Memory)

In the previous algorithms we treated the air quality prediction as a regression problem with features and labels. Here in LSTM we treat the problem as a univariate time series problem.

We try to model the seasonality and trend of air quality in the different time intervals of the data like daily and hourly. Air quality is a very hard problem to model on the basis of parameters since there are so many factors associated but training the data in seasonality yields a better result for us to make somewhat accurate assumptions about future quality of year. We can study if the overall air quality will go down or the real predicted value for the air quality using LSTMs.

In LSTM our entire dataset structure is changed and now it being a univariate time series we use only pm2.5 as our feature and the value to predict.

We start from the beginning a new machine learning pipeline to implement LSTM. The implementation methodology is described by the following pipeline diagram:

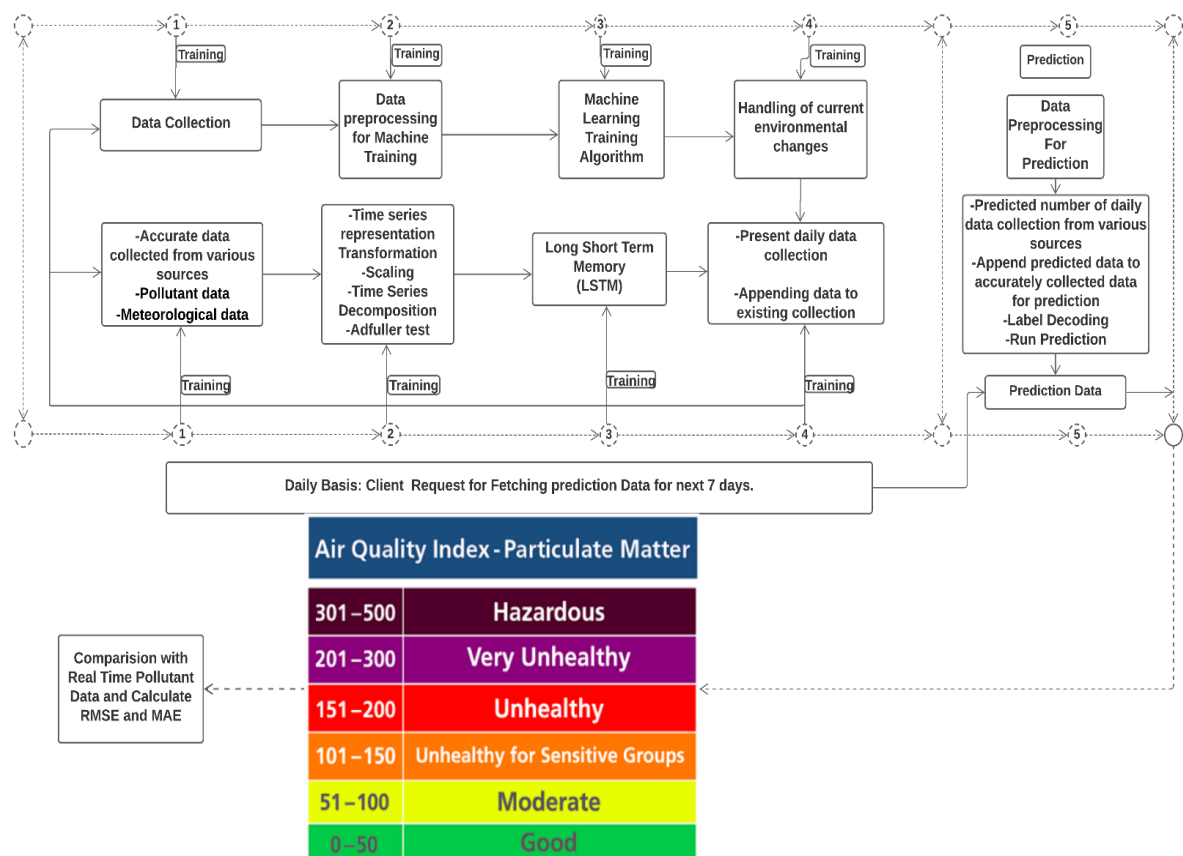


Figure 5.1 LSTM Implementation Pipeline

## Step 1: Data collection

- Data is collected from AQICN website for US embassy station
- Data was extracted from the website with a frequency of daily every 10 days

## Step 2: Data Analysis

### 2.1 Load the dataset using pandas

```
data =  
pd.read_csv('us_embassy_kathmandu_upto_aug_31_pm25_only_sorted.csv',  
            Parse_dates = True, index_col = 'date')
```

### 2.2 Check for null values

```
data.isnull().sum()
```

If null values, go to step 2.2.1

2.2.1 handle missing values using forward fill method of imputation

Else, go to step 2.3

### 2.3 Plot the data

### 2.4 Decompose the data into seasonality, trend and residuals using statsmodel

```
seas = sm.tsa.seasonal_decompose(data.values, model='additive',  
period=30)  
seas.plot()
```

### 2.5 Perform adfuller test on the data

```
result = adfuller(data)
```

## Step 3: Preprocess the data

### 3.1 check if the data is stationary or not using the result of adfuller test,

### 3.2 Create baseline error statistics using lag-1 prediction

#### 3.2.1 create lag-1 predictions dataframe

```
lag_1_data = pd.concat(data, data.shift(1), axis=1)
```

```
lag_1_data = lag_1_data[1:]
```

```
Lag_1_data.columns = ['original_pm25', 'predicted_pm25']
```



### 3.2.2 Calculate base-line errors

```
rmse_baseline=np.sqrt(mean_squared_error(lag_1_data.original_pm25,
lag_1_data.predicted_pm25))
```

```
mae_baseline = mean_absolute_error(lag_1_data.original_pm25,
lag_1_data.predicted_pm25)
```

### 3.3 Perform scaling on the data

```
scaler = MinMaxScaler(feature_range = (0,1))
```

```
data = scaler.fit_transform(np.array(data).reshape(-1,1))
```

### 3.4 Split the data into train and test set

- A 75-25 split was done

### 3.5 Convert the univariate dataset into supervised learning problem

- N lag steps are used, the dataset will be in the form  $X = t, t+1, t+3$  and  $Y = t+4$

```
X_train, y_train = create_dataset(train_data, time_step)
```

```
X_test, y_test = create_dataset(test_data, time_step)
```

## Step 4: Model Selection

### 4.1 import necessary models from tensorflow keras api

### 4.2 Select the appropriate model

- minimum of two LSTM layers should be selected
- A final dense layer with single neuron to get output
- Use convolutions if necessary
- optionally use a 1d convolutions to extract features in auto correlated data

### 4.3 Print model summary

```
print(model.summary())
```

### 4.4 Compile the model

```
Learning rate = 0.01
```

```
Optimizer = adam
```

```
Loss function = mean_squared_error
```

## Step 5: Model training

```
model.fit(X_train,y_train, validation_data=(X_test, y_test), epochs=150, verbose=1)
```

Step 6: Make predictions

```
train_predict = model.predict(X_train)
```

```
test_predict = model.predict(X_test)
```

Step 7: Perform error statistics in training and test sets

- MAE and RMSE will be calculated and compared against baseline error
- Plot the predictions against real values

Step 8: Make predictions into the future

Perform inverse scaling in the predicted values

Step 9: Export

9.1 Export the predicted values into a csv file

```
lstm_output.to_csv('sept_predictions.csv')
```

9.2 Add the predicted values to the original dataset

```
data.extend(lstm_output)
```

9.3 Export the model

## 5.2 Analysis

The trained model was analyzed on different algorithms to get the AQI and validation scores were compared on different evaluation metrics. The evaluation metrics used were mean absolute error and root mean square error.

Two measures of AQI namely ozone and particulate matter 2.5 were predicted using different parameters.

### 5.2.1 Multi linear Regression

*Table 5.1: MLR Prediction Analysis*

| Pollutants | Iterator | n-fold | MAE    | RMSE   |
|------------|----------|--------|--------|--------|
| PM2.5      | 50       | 5      | 13.38  | 52.41  |
|            | 100      | 5      | 16.67  | 26.21  |
|            | 150      | 5      | 17.06  | 27.55  |
|            | 200      | 5      | 17.14  | 28.82  |
|            | 50       | 10     | 12.816 | 17.974 |
|            | 100      | 10     | 17.806 | 24.787 |
|            | 150      | 10     | 12.633 | 18.111 |
|            | 200      | 10     | 23.788 | 31.782 |

### 5.2.2 Random Forest Regression

*Table 5.2: RFR Prediction Analysis*

| Pollutants | Tree Depth | Iterations | n-fold | MAE    | RMSE   |
|------------|------------|------------|--------|--------|--------|
| PM 2.5     | 10         | 50         | 5      | 12.699 | 17.081 |
|            | 10         | 100        | 5      | 12.705 | 17.103 |
|            | 10         | 150        | 5      | 12.695 | 17.090 |
|            | 10         | 200        | 5      | 12.698 | 17.114 |
|            | 10         | 50         | 10     | 12.701 | 17.099 |
|            | 10         | 100        | 10     | 12.704 | 17.121 |
|            | 10         | 150        | 10     | 12.684 | 17.088 |

|  |    |     |    |        |        |
|--|----|-----|----|--------|--------|
|  | 10 | 200 | 10 | 12.651 | 17.091 |
|  | 5  | 50  | 10 | 12.680 | 16.964 |
|  | 5  | 100 | 10 | 12.625 | 16.899 |
|  | 5  | 150 | 10 | 12.604 | 16.886 |
|  | 5  | 200 | 10 | 12.609 | 16.894 |
|  | 5  | 50  | 5  | 12.654 | 17.076 |
|  | 5  | 100 | 5  | 12.705 | 17.162 |
|  | 5  | 150 | 5  | 12.571 | 16.821 |
|  | 5  | 200 | 5  | 12.656 | 16.917 |
|  | 3  | 10  | 10 | 12.604 | 16.896 |



Figure 5.2 Real vs Predicted Values Train set using RFR

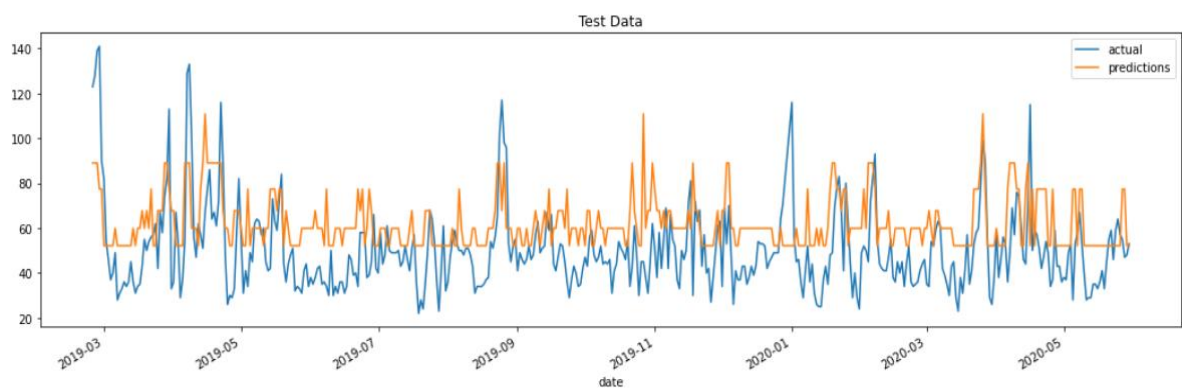


Figure 5.3 Real vs Predicted values Test set using RFR

### 5.2.3 LTSM Analysis

*Trend - Seasonality-Residual Decomposition:*

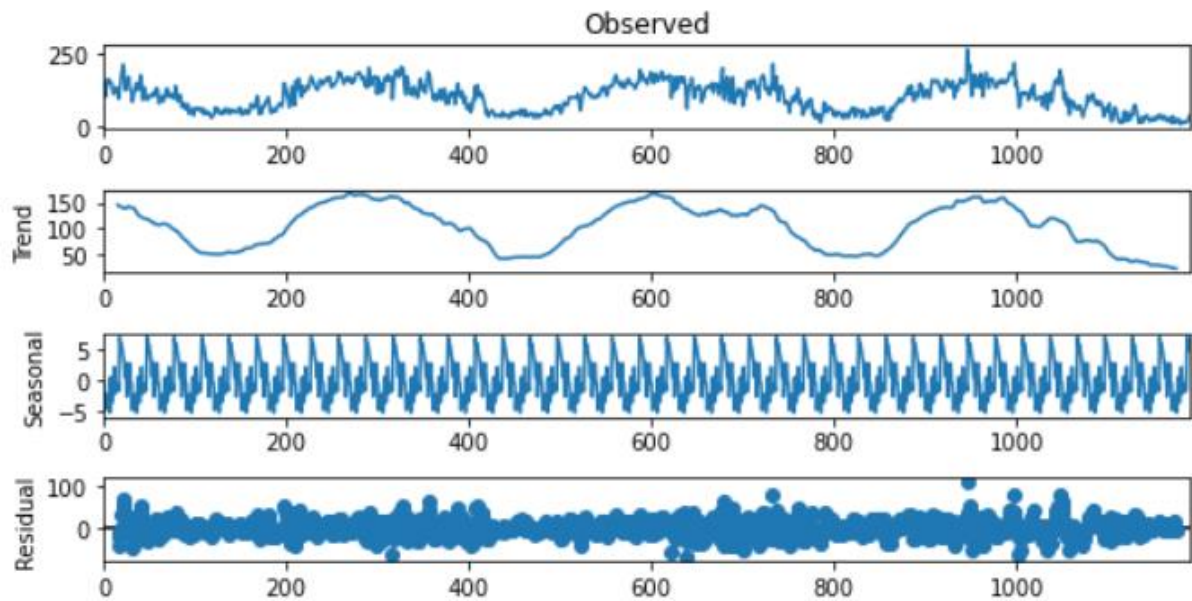


Figure 5.4 Decomposition of PM2.5 into trend, seasonality and residuals

Observation:

- There is a clear seasonal pattern in the data
- There is no clear upwards or downwards trend but a constant pattern in the data
- Almost all the particulate matter concentration can be modelled by the residuals

*Ad fuller test:*

H0(Null hypothesis): Data is non-stationary

H1(Alternative Hypothesis): Data is stationary

ADF Statistic: -2.285690

p-value: 0.176625

Critical Values:

1%: -3.436

5%: -2.864

10%: -2.568

Observation:

- Since  $-2.285 > -3.436, -2.864, -2.568$  (t-values at 1 %, 5% and 10% confidence interval), we cannot reject the null hypothesis. Hence data is non-stationary (which means data has relation to time)
- $p\text{-value} > 0.05$ : Fail to reject the null hypothesis (H0), the data is non-stationary and has time dependent structure

Table 5.3: LSTM Prediction Analysis

| Epochs | Time step | loss   | RMSE   | MAE    |
|--------|-----------|--------|--------|--------|
| 50     | 7         | 0.0044 | 16.839 | 12.074 |
|        | 30        | 0.0046 | 17.912 | 12.851 |
|        | 100       | 0.0045 | 17.201 | 12.176 |
| 100    | 7         | 0.0043 | 16.827 | 12.029 |
|        | 30        | 0.0041 | 15.503 | 11.156 |
|        | 100       | 0.0041 | 16.302 | 11.639 |
| 150    | 7         | 0.0038 | 15.691 | 11.320 |
|        | 30        | 0.0037 | 15.503 | 11.156 |
|        | 100       | 0.0041 | 16.469 | 12.068 |
| 200    | 7         | 0.0042 | 17.657 | 12.625 |
|        | 30        | 0.0034 | 18.319 | 12.797 |
|        | 100       | 0.0036 | 18.893 | 13.410 |

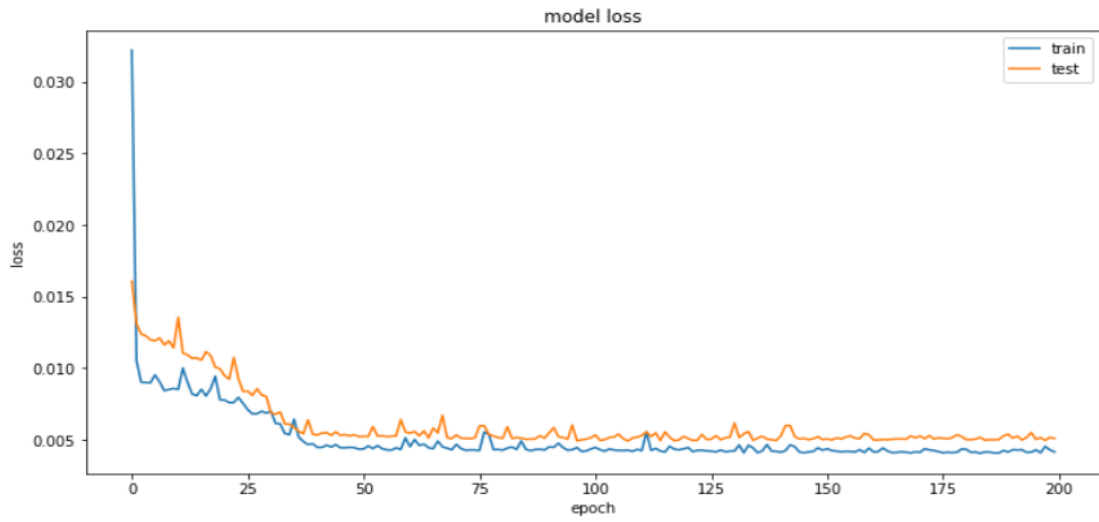
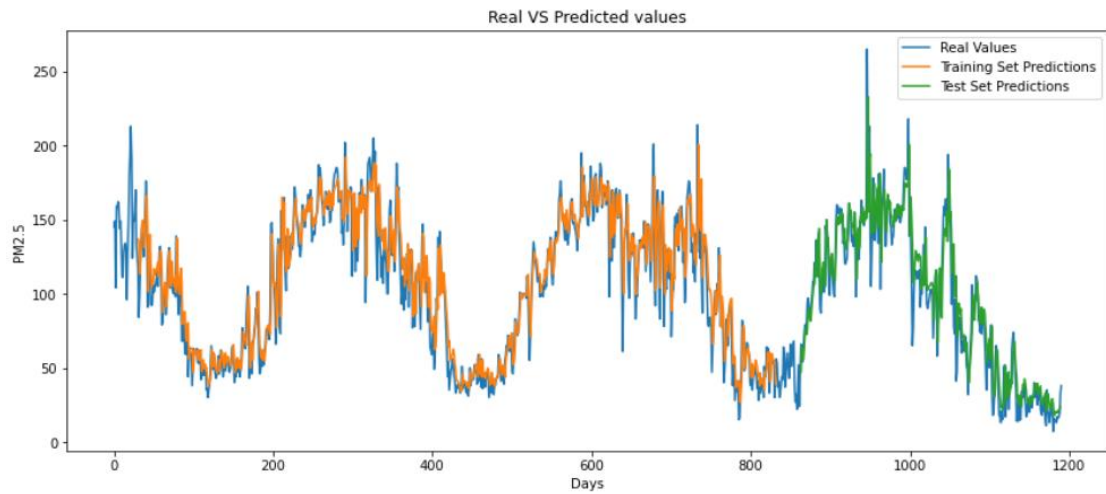


Figure 5.5 Training vs Validation Loss



*Figure 5.6 Real vs Predicted values using LSTM*

#### 5.2.4 Comparative analysis:

*Table 5.4: Comparison of results of all algorithms*

| Algorithm                | RMSE   | MAE    |
|--------------------------|--------|--------|
| Multilinear Regression   | 28.455 | 16.411 |
| Random Forest Regression | 18.081 | 13.452 |
| LSTM                     | 16.926 | 12.108 |

Thorough comparative analysis we select LSTM for deploying into the website due to following reasons:

- LSTM is adaptable to new readings daily
- LSTM can scalable to predict hourly, daily and weekly values
- LSTM can be used on a single feature set
- LSTM required less time for data preprocessing

### Post Deployment Results:

After LSTM was deployed, we tested its efficiency by predicting AQI for the month of August, 2020 at an interval of 10 days. We took predicted values for 10 days and real values and compared them. We also retrained the algorithm every 10 days by integrating new readings to the existing data. We considered 10 as the acceptable error margin, so absolute value of the difference predicted and real values should be equal or less than the error margin. The comparison of the prediction and real values is given below:

*Table 5.5: Comparison of Real AQI and Predicted AQI after Deploying*

| Date       | Predict<br>ed<br>values | Real<br>Values | Predict<br>ed<br>Group | Real<br>Group | Error | Acceptance | AQI<br>accuracy | Group<br>Accurac<br>y |
|------------|-------------------------|----------------|------------------------|---------------|-------|------------|-----------------|-----------------------|
| 01/08/2020 | 34                      | 25             | Good                   | Good          | 9     | TRUE       | 80.00%          | 100.00%               |
| 02/08/2020 | 34                      | 29             | Good                   | Good          | 5     | TRUE       |                 |                       |
| 03/08/2020 | 33                      | 32             | Good                   | Good          | 1     | TRUE       |                 |                       |
| 04/08/2020 | 32                      | 36             | Good                   | Good          | 4     | TRUE       |                 |                       |
| 05/08/2020 | 32                      | 22             | Good                   | Good          | 10    | TRUE       |                 |                       |
| 06/08/2020 | 31                      | 19             | Good                   | Good          | 12    | FALSE      |                 |                       |
| 07/08/2020 | 32                      | 18             | Good                   | Good          | 14    | FALSE      |                 |                       |
| 08/08/2020 | 32                      | 26             | Good                   | Good          | 6     | TRUE       |                 |                       |
| 09/08/2020 | 30                      | 21             | Good                   | Good          | 9     | TRUE       |                 |                       |
| 10/08/2020 | 32                      | 31             | Good                   | Good          | 1     | TRUE       |                 |                       |
|            |                         |                |                        |               |       |            |                 |                       |
| 11/08/2020 | 17                      | 15             | Good                   | Good          | 2     | TRUE       | 90.00%          | 100.00%               |
| 12/08/2020 | 19                      | 11             | Good                   | Good          | 8     | TRUE       |                 |                       |
| 13/08/2020 | 19                      | 35             | Good                   | Good          | 16    | FALSE      |                 |                       |



|            |    |    |      |      |    |       |        |         |
|------------|----|----|------|------|----|-------|--------|---------|
| 14/08/2020 | 20 | 22 | Good | Good | 2  | TRUE  |        |         |
| 15/08/2020 | 20 | 19 | Good | Good | 1  | TRUE  |        |         |
| 16/08/2020 | 20 | 13 | Good | Good | 7  | TRUE  |        |         |
| 17/08/2020 | 20 | 18 | Good | Good | 2  | TRUE  |        |         |
| 18/08/2020 | 20 | 23 | Good | Good | 3  | TRUE  |        |         |
| 19/08/2020 | 20 | 26 | Good | Good | 6  | TRUE  |        |         |
| 20/08/2020 | 20 | 17 | Good | Good | 3  | TRUE  |        |         |
|            |    |    |      |      |    |       |        |         |
| 21/08/2020 | 22 | 7  | Good | Good | 15 | FALSE | 80.00% | 100.00% |
| 22/08/2020 | 24 | 16 | Good | Good | 8  | TRUE  |        |         |
| 23/08/2020 | 23 | 15 | Good | Good | 8  | TRUE  |        |         |
| 24/08/2020 | 24 | 14 | Good | Good | 10 | TRUE  |        |         |
| 25/08/2020 | 22 | 13 | Good | Good | 9  | TRUE  |        |         |
| 26/08/2020 | 24 | 17 | Good | Good | 7  | TRUE  |        |         |
| 27/08/2020 | 21 | 16 | Good | Good | 5  | TRUE  |        |         |
| 28/08/2020 | 24 | 17 | Good | Good | 7  | TRUE  |        |         |
| 29/08/2020 | 24 | 19 | Good | Good | 5  | TRUE  |        |         |
| 30/08/2020 | 24 | 33 | Good | Good | 9  | TRUE  |        |         |
| 31/08/2020 | 24 | 30 | Good | Good | 6  | TRUE  |        |         |
|            |    |    |      |      |    |       |        |         |

|            |    |    |      |      |    |       |        |         |
|------------|----|----|------|------|----|-------|--------|---------|
| 01/09/2020 | 29 | 38 | Good | Good | 9  | TRUE  | 80.00% | 100.00% |
| 02/09/2020 | 28 | 27 | Good | Good | 1  | TRUE  |        |         |
| 03/09/2020 | 28 | 18 | Good | Good | 10 | TRUE  |        |         |
| 04/09/2020 | 28 | 24 | Good | Good | 4  | TRUE  |        |         |
| 05/09/2020 | 30 | 40 | Good | Good | 10 | TRUE  |        |         |
| 06/09/2020 | 32 | 42 | Good | Good | 10 | TRUE  |        |         |
| 07/09/2020 | 32 | 48 | Good | Good | 16 | FALSE |        |         |
| 08/09/2020 | 31 | 39 | Good | Good | 8  | TRUE  |        |         |
| 09/09/2020 | 30 | 40 | Good | Good | 10 | TRUE  |        |         |
| 10/09/2020 | 29 | 41 | Good | Good | 12 | FALSE |        |         |

## 5.3 Tools Used

### 5.3.1 Analysis and Design Tools

- Creately**  
 Creately is a dynamic diagramming tool that can be deployed from the cloud or can be used as a desktop application. Creately has been used for drawing of use-case diagrams, sequence diagram and use case diagrams of the project.
- Figma**  
 Figma is a vector graphics editor and prototyping tool. Primarily online, figma provides with tools needed for the design phase of a project including vector tools which are capable of fully fledged illustration as well as prototyping capabilities and code generation. Figma has been used for designing the interface of the project website.

### 5.3.2 Implementation Tools

- Python**  
 Python is an interpreted, high-level, general-purpose programming language. Python has been gaining popularity as a programming language for machine learning.

So, python has been used as the main programming language for our Air Quality Prediction project.

- Jupyter Notebook

Jupyter Notebook is a web application that allows users to create and share documents that contain live code, equations, visualizations and narrative texts. Jupyter Notebook is mostly sought-after framework by machine learning enthusiasts for its ease of use feature.

Jupyter Notebook has been used in our project for testing our ML codes and making different visualizations related to air quality data.

- Django

Django is an open source web application framework based on python and follows MVC architectural pattern.

Django has been used for development of our project website since it uses Python programming language and allows rapid development of sites as it offers a complete suite of features than other web frameworks.

- Google colab

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

Colab has been used in the project as it provides free GPU to run our model with large datasets in the browser without needing much higher hardware requirement.

- VS Code

VS Code is a free code editor developed by Microsoft. It is a lightweight but powerful source code editor which runs on desktop and is available for Windows, macOS and Linux.

VS Code has been used as our main code editor to write code for our project mainly for the website frontend.

- Mapbox API

Mapbox is an open source mapping platform for custom designed maps. Mapbox is a developer platform used across industries to create custom applications that solve problems with maps, data, and spatial analysis. It provides building blocks to add location features like maps, search, and navigation into any experience you create. Mapbox API has been used in our project to pinpoint the location (Kathmandu in our case) to show the AQI level of Kathmandu.

- Airnow api

Airnow api is used to display real time weather data and real time air quality data in the homepage

## 5.4 Testing

The testing of a complete and fully integrated software to find out bugs by executing software in different instances is called Software testing. The sole purpose of this process is to exercise the system by validating and verifying the system. Different inputs are given to check if the application gives us expected outputs.

Testing is one of the most important processes that should be performed during the development process. Testing is a process that spans throughout the Software development life cycle (SDLC). Testing can be considered a set of activities that expose system and vulnerabilities and fix them

### 5.4.1 Unit Testing:

Unit Testing is a level of software testing where individual units/components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of a software. It usually consists of one or few units and usually a single unit. We tested the modules after coding it hence; the unit testing was done concurrently and made bug free after encountering them

*Table 5.6: Test Case for different Units*

| Test Case                                      | Test Input   | Test Result                  |
|--|--|------------------------------|
| Creating dataset before feeding to the network | List input of various lengths                                      | Dataset created successfully |
| Sign up  | Different combinations of username and password                    | User created successfully    |
| Login  | Different combination of username and password of registered users | User logged in successfully  |

### 5.4.2 Integration Testing:

The main focus of this testing is to check the data communication between different modules. It combines the individual components of the system and checks if for any error.

## Test Case 1

*Table 5.7: Test Case for Prediction Module*

| Steps | Test Steps               | Test Data  | Expected Result                                     | Actual Results                                     | Status |
|-------|--------------------------|------------|---|--|--------|
| 1     | Open the prediction page |            | Prediction page should open without error and login | Prediction page appears                            | Pass   |
| 2     | Select past date         | 2020-07-07 | User should not be able to select past dates        | Past date is greyed out and cannot be selected     | Pass   |
| 3     | Select future date       | 2020-09-10 | Future date should be selected                      | Future date is selected and ready to be predicted  | Pass   |
| 4     | Click predict button     |            | Prediction for future date must be displayed        | Predicted AQI along with AQI category is displayed | Pass   |

## Test Case 2

*Table 5.8: Test Case for Past Data Module*

| Steps | Test Steps                     | Test Data | Expected Result  | Actual Results  | Status |
|-------|--------------------------------|-----------|--|---|--------|
| 1     | Navigate to the past data page |           | Past date should open without error and login            | Past date with year selector opened                           | Pass   |
| 2     | Select the year                | 2017      | 2017's historical data page should be opened             | Data summary and plot of data was displayed                   | Pass   |
| 3     | Select the graph               | 2019      | 2019's graph on different parameters should be displayed | Graph of the data for 2019 was shown for different parameters | Pass   |
| 4     | Select download now button     |           | Entire dataset should be downloaded                      | Entire dataset is downloaded in csv format                    | Pass   |

## CHAPTER 6: CONCLUSION AND FUTURE WORKS

### 6.1 Conclusion

In recent years, air pollution has been a major issue for Kathmandu, and lots of advanced models for forecasting air pollutants have been proposed. In this study, we evaluated models based on traditional method i.e. Multilinear Regression (MLR), modern machine learning methods like Random Forest Regression (RFR) and Long Short Term Memory Systems (LSTMs) to predict Air Quality Index based on the pollutant PM<sub>2.5</sub>. We treated the prediction problem from two ways, first by treating the problem as a conventional machine learning problem with features and labels. The features were meteorological and air pollutant factors combined and the label was PM<sub>2.5</sub>. This parameterized problem was fed to MLR and RFR. The other approach we took was to treat the problem as a time series problem and model the temporal dependencies in the data. The univariate time series data was fed into stacked LSTM to make predictions

The experimental results of each of our approaches are shown in the above tables and based on those results we selected LSTM as our method of prediction since, this method showed high precision, long range prediction and strong adaptive ability. The results show that deep learning-based models clearly outperformed the conventional statistical models. Furthermore, it showed that with enough modeling parameters RFR can be used to accurately predict the AQI without overfitting. LSTM provided a wide range of generalization by providing a good approximation of non-linearly mapped as well which the MLR failed to accomplish. series

The stacked LSTM model is a successful attempt and will be a promising start for air pollution concentration forecasting. This model can be used in other multivariate input time series prediction problems and also expanded to include other pollutant parameters.

### 6.2 Future Works

This project lays basic foundation to the prediction of AQI based on a single pollutant concentration but it can be further expanded to include other pollutants such as Ozone, PM<sub>10</sub>, SO<sub>2</sub>, CO, CO<sub>2</sub> etc.

The concentration of pollution is restricted by any factors hence the prediction accuracy may be inaccurate when other factors showed greater impact. Therefore, we suggest using more explanatory variables to increase performance. Incorporating other factors such as geographical, meteorological, traffic concentration etc. along with suitable prediction models will yield better accuracy.

Furthermore, we only deployed the predictions at daily intervals but this project can be expanded to include predictions on hourly, weekly and monthly basis. Our experiments indicate significantly better performance at hourly intervals.

A single station's readings were used for this research whereas in the future multiple sensor locations can be used to make predictions for wider coverage and better accuracy. In the future we can expand this research to come up with a prediction method for non-sensor locations.

## REFERENCES

|      |  |
|------|--|
| [1]  | B. Saud and G. Paudel, "The Threat of Ambient Air Pollution in Kathmandu, Nepal," <i>Journal of Environmental and Public Health</i> , vol. 2018, pp. 1–7, Oct. 2018.   |
| [2]  | J. Yoo, D. Shin, and D. Shin, "Prediction System for Fine Particulate Matter Concentration Index by Meteorological and Air Pollution Material Factors Based on Machine Learning," <i>Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019</i> , 2019.   |
| [3]  | B. Liu, C. Shi, J. Li, Y. Li, J. Lang, and R. Gu, "Comparison of Different Machine Learning Methods to Forecast Air Quality Index," <i>Lecture Notes in Electrical Engineering Frontier Computing</i> , pp. 235–245, 2019.   |
| [4]  | Saba Fotouhi, M. Hassan Shirali-Shahreza, and Adel Mohammadpour. 2018. Concentration Prediction of Air Pollutants in Tehran. In Proceedings of the international conference on smart cities and internet of things (SCIOT '18). Association for Computing Machinery, New York, NY, USA, Article 7, 1–7. DOI: <a href="https://doi.org/10.1145/3269961.3269967">https://doi.org/10.1145/3269961.3269967</a> |
| [5]  | P. Hajek and V. Olej, "Predicting Common Air Quality Index – The Case of Czech Microregions," <i>Aerosol and Air Quality Research</i> , vol. 15, no. 2, pp. 544–555, 2015.   |
| [6]  | Bai, Y., Zeng, B., Li, C., & Zhang, J. (2019). An ensemble long short-term memory neural network for hourly PM2.5 concentration forecasting. <i>Chemosphere</i> , 222, 286-294. doi:10.1016/j.chemosphere.2019.01.121  |
| [7]  | Bansal, Mohit & Aggarwal, Anirudh & Verma, Tanishq & Sood, Apoorvi. (2019). Air Quality Index Prediction of Delhi using LSTM. 10.13140/RG.2.2.26885.70884.   |
| [8]  | Chaudhary, V., Deshbhratar, A., Kumar, V., Paul, D., & Samsung (2018). Time Series Based LSTM Model to Predict Air Pollutant's Concentration for Prominent Cities in India.  |
| [9]  | Karimian, H., Li, Q., Wu, C., Qi, Y., Mo, Y., Chen, G., . . . Sachdeva, S. (2019). Evaluation of Different Machine Learning Approaches to Forecasting PM2.5 Mass Concentrations. <i>Aerosol and Air Quality Research</i> , 19(6), 1400-1410. doi:10.4209/aaqr.2018.12.0450   |
| [10] | Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. <i>Neural computation</i> . 9. 1735-80. 10.1162/neco.1997.9.8.1735.  |
| [11] | Jiao, Y., Wang, Z., & Zhang, Y. (2019). Prediction of Air Quality Index Based on LSTM. <i>2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)</i> . doi:10.1109/itaic.2019.8785602   |
| [12] | Qin, Z., Cen, C., & Guo, X. (2019). Prediction of Air Quality Based on KNN-LSTM. <i>Journal of Physics: Conference Series</i> , 1237, 042030. doi:10.1088/1742-6596/1237/4/042030  |

## APPENDIX

### Model Summary

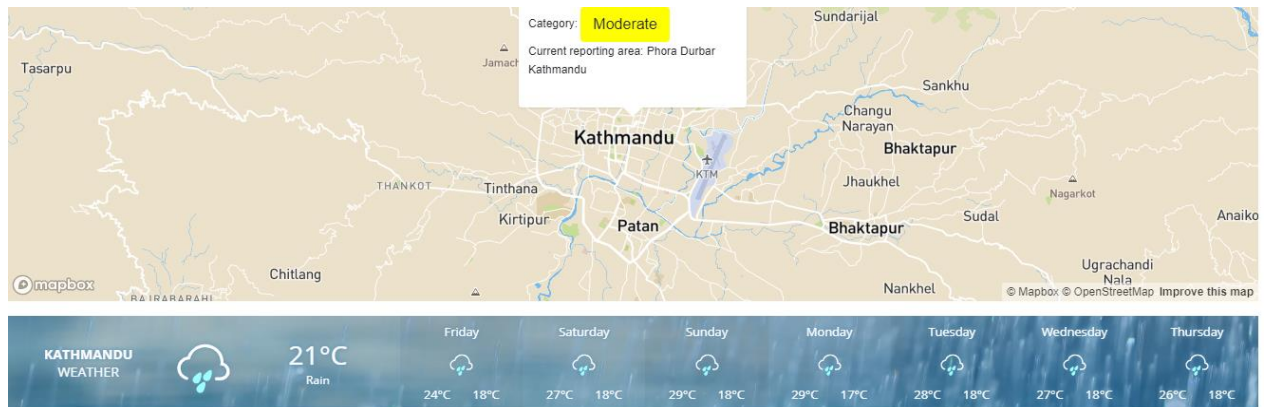
Model: "sequential\_1"

| Layer (type)             | Output Shape    | Param # |
|--------------------------|-----------------|---------|
| =====                    |                 |         |
| =                        |                 |         |
| conv1d_1 (Conv1D)        | (None, 300, 60) | 360     |
| <hr/>                    |                 |         |
| lstm_1 (LSTM)            | (None, 300, 60) | 29040   |
| <hr/>                    |                 |         |
| lstm_2 (LSTM)            | (None, 300, 60) | 29040   |
| <hr/>                    |                 |         |
| lstm_3 (LSTM)            | (None, 60)      | 29040   |
| <hr/>                    |                 |         |
| dense_1 (Dense)          | (None, 30)      | 1830    |
| <hr/>                    |                 |         |
| dense_2 (Dense)          | (None, 10)      | 310     |
| <hr/>                    |                 |         |
| dense_3 (Dense)          | (None, 1)       | 11      |
| =====                    |                 |         |
| =                        |                 |         |
| Total params: 89,631     |                 |         |
| Trainable params: 89,631 |                 |         |
| Non-trainable params: 0  |                 |         |

GitHub Link for Entire project: <https://github.com/Basantakhadka/Air-Quality-Prediction>



## Screenshots



### O<sub>3</sub> Current AQI : 29

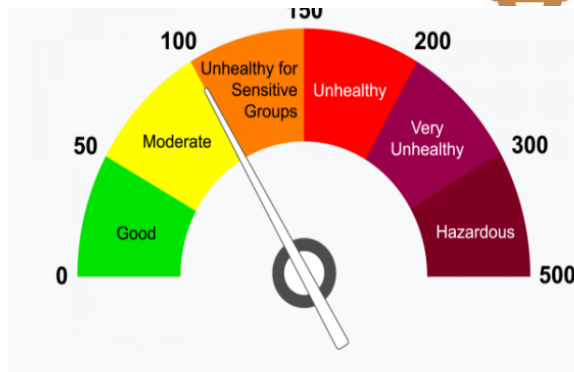
(0-50) Air quality is satisfactory, and air pollution poses little or no risk.. Current reporting area is Phora Durbar Kathmandu

Good

### PM2.5 Current AQI : 66

(51-100) Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.. Current reporting area is Phora Durbar Kathmandu

Moderate



### Category of AQI

Generally, there are six air quality index category : Good , Moderate , Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy and Harzardous. These are represented by particular colors.....


[Read More](#)

| PROJECT  | USEFUL LINKS  | CONTACT   |
|--|---|---|
| AQI stands for air quality index. Air quality index (AQI) is used by government agencies to communicate to the public how polluted the air currently is or how polluted it's forecast. | <a href="#">Home</a><br><a href="#">About</a><br><a href="#">Predict</a><br><a href="#">Past_data</a> | kathmandu, 44600, Nepal<br><a href="mailto:rayriderkhadka5@gmail.com">rayriderkhadka5@gmail.com</a><br><a href="mailto:sapkotaumesh4@gmail.com">sapkotaumesh4@gmail.com</a><br><a href="mailto:utsavdh@gmail.com">utsavdh@gmail.com</a><br><a href="mailto:nbgc7@gmail.com">nbgc7@gmail.com</a> |

## Homepage

The screenshot shows the homepage of the 'Air Quality Prediction Kathmandu' website. The header includes the site name and navigation links: Home, About, Predict, Past Data, and Contact us. The main content area features a 'Please Sign in' section with input fields for 'Username' and 'Password', and a blue 'Sign in' button.

## Login page


**Air Quality Prediction Kathmandu**

[Home](#)
[About](#)
[Predict](#)
[Past Data](#)
[Contact us](#)

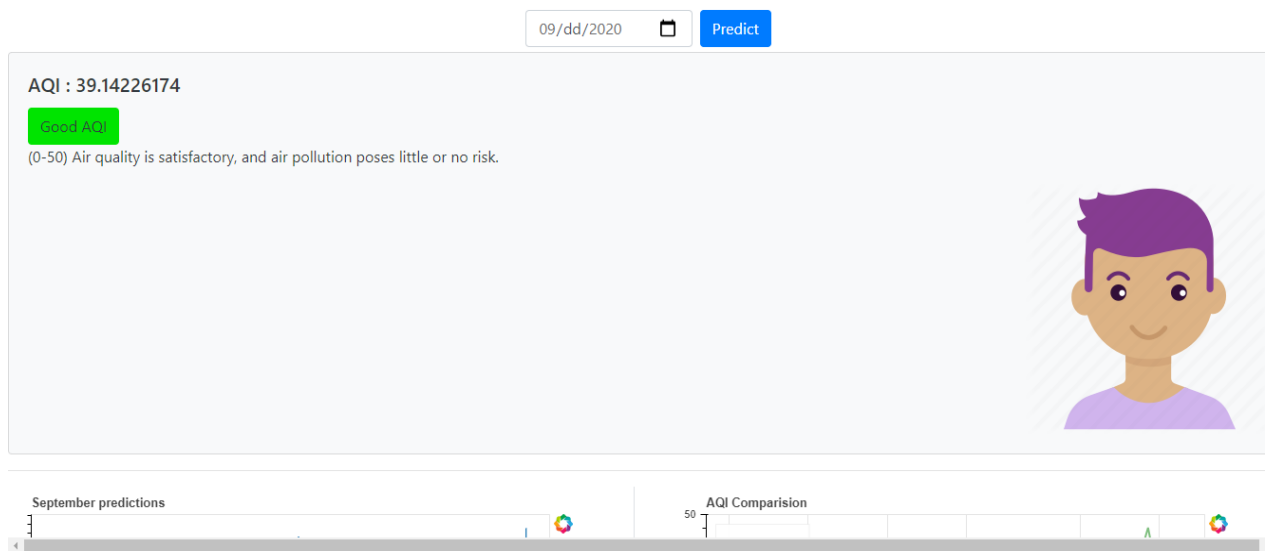
### Please register


[Sign up](#)

If you already have an account? [login](#)


## Register page

### Kathmandu AQI Prediction




**Air Quality Prediction Kathmandu**






[Home](#)
[About](#)
[Predict](#)
[Past Data](#)
[Contact us](#)



#### Kathmandu AirQuality

AQI stands for air quality index. Air quality index (AQI) is used by government agencies to communicate to the public how polluted the air currently is or how polluted it is forecast to become. Public health risks increase as the AQI rises.

In the summer, Kathmandu's PM2.5 falls to very close to the WHO annual limit. But in the winter, it shoots up to as high as 9 times that amount. And remember, that's the average each month. Certain days will be much higher than that. It happens because summer weather patterns push more pollution away and because winter brings winter heating.

## About page

[2017](#)
[2018](#)
[2019](#)
[2020](#)

### PROJECT

AQI stands for air quality index. Air quality index (AQI) is used by government agencies to communicate to the public how polluted the air currently is or how polluted it's forecast.

TOP

### USEFUL LINKS

[Home](#)  
[About](#)  
[Predict](#)  
[Past\\_data](#)  
[Download dataset](#)

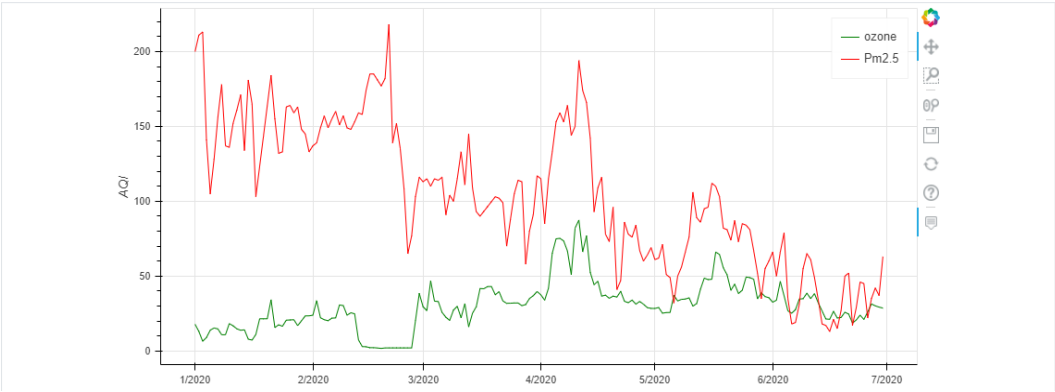
### CONTACT

kathmandu, 44600, Nepal

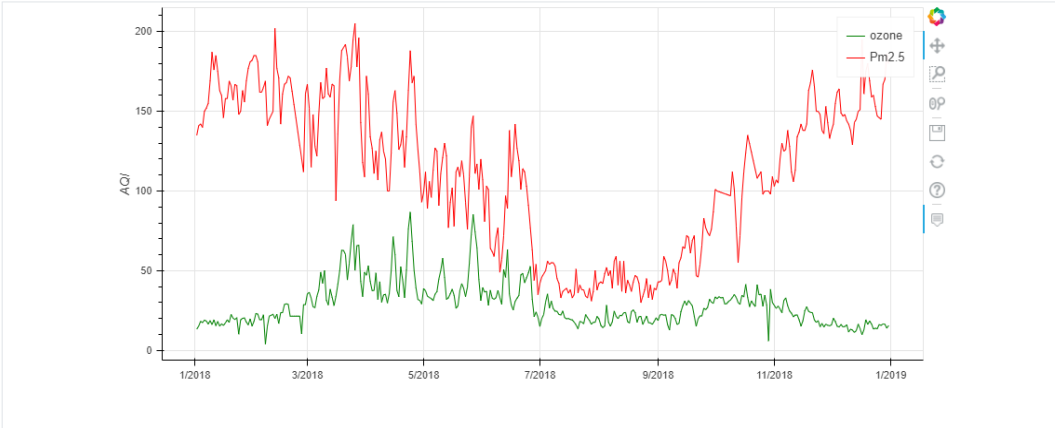
[rayriderkhadka5@gmail.com](mailto:rayriderkhadka5@gmail.com)  
[sapkotaumesh4@gmail.com](mailto:sapkotaumesh4@gmail.com)  
[utsavdh@gmail.com](mailto:utsavdh@gmail.com)  
[nbngc7@gmail.com](mailto:nbngc7@gmail.com)

© 2020 Copyright: **Airqualityprediction**

[2017](#)
[2018](#)
[2019](#)
[2020](#)



[2017](#)
[2018](#)
[2019](#)
[2020](#)



Past data page

| PROJECT  | USEFUL LINKS                     | CONTACT  |
|--|----------------------------------|--|
| AQI stands for air quality index. Air quality index (AQI) is used by government agencies to communicate to the public how polluted the air currently is or how polluted it's forecast. | <a href="#">Home</a>             | kathmandu, 44600, Nepal  |
|  | <a href="#">About</a>            | <a href="mailto:rayriderkhadka5@gmail.com">rayriderkhadka5@gmail.com</a> |
|  | <a href="#">Predict</a>          | <a href="mailto:sapkotaumesh4@gmail.com">sapkotaumesh4@gmail.com</a>     |
|  | <a href="#">Past_data</a>        | <a href="mailto:utsavdh@gmail.com">utsavdh@gmail.com</a>                 |
| <a href="#">TOP</a>  | <a href="#">Download dataset</a> | <a href="mailto:nbngc7@gmail.com">nbngc7@gmail.com</a>                   |

127.0.0.1:8000/about

© 2020 Copyright: **Airqualityprediction**

## Contact page

**The End**