AirQo Data Analysis Report

Abdul-Ganiy B. Usman

usman.aganiy@gmail.com

1 Introduction

This report present the analysis on air quality data collected since 2019 to date from 42 different AirQo monitoring sites.

2 Background

This section describes briefly the implementation and analysis approach of the dataset.

2.1 Data

The data provided contain over 2 million data point with 7 features, namely: *channel_id, pm2_5, pm10, s2_pm2_5, s2_pm10, Site* and *TimeStamp*. The *channel_id* is a unique identifier for each *Site*, which is a column containing the location name. The *pm2_5* and *s2_pm2_5* columns contain **particulate matter 2.5** readings for sensors one and two respectively, same for *pm10* and *s2_pm10* columns for **particulate matter 10**. The data also comes with a few metadata about each site.

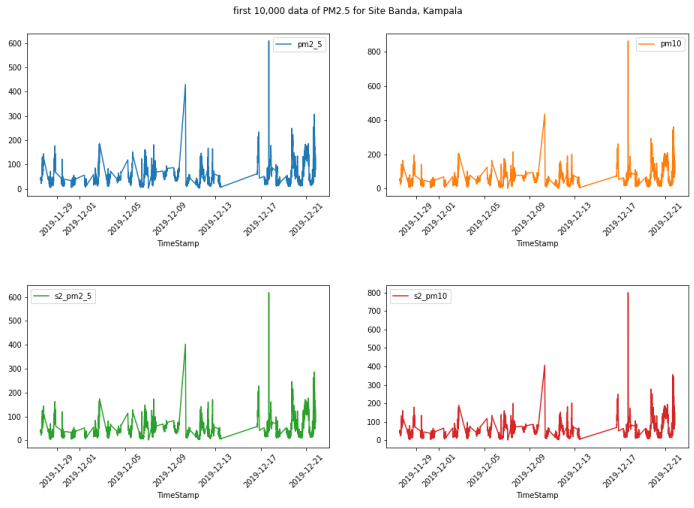


Figure 1: Input window

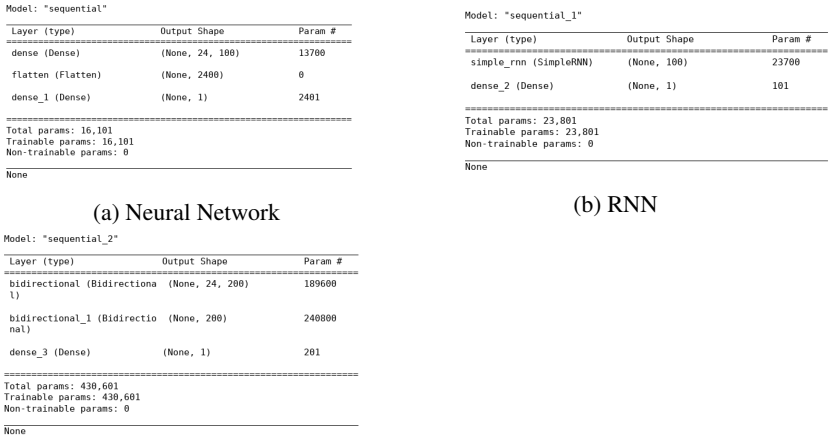
2.2 feature engineering

Before diving in to build a model, it is highly essential to understand the data and be certain one is passing the model appropriately formatted data. In our case, where there is only the time data to predict the air quality values for the two sensors, we leverage largely on the *TimeStamp* data to engineer more features following the steps below:

First, we created hourly timeseries data by average all data readings over the hourly time steps. Then, we extracted from the series the following time data, $\{hour \in [0, 23]\}$, $\{day \in [1, 31]\}$, $\{day\_of\_week \in [1, 7]\}$, $\{month \in [1, 12]\}$ $\{week\_of\_year \in [1, 53]\}$. Also, a one-hot encoding of each of the time data was added to the features along with their *sine* and *cosine* values. These sine and cosine are important to preserve the seasonality or cyclical nature of our data. Lastly, the *particulate matter* readings are included to the feature list since the goal is to use past values to predict the future.

2.3 model setup

The models considered for the analysis are Neural Network (NN), Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM). These models have been carefully selected as they are more suitable for our analysis problem, timeseries forecasting. The models architecture are presented below:



(a) Neural Network

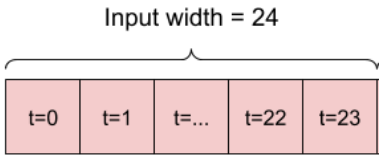
(b) RNN

(c) LSTM

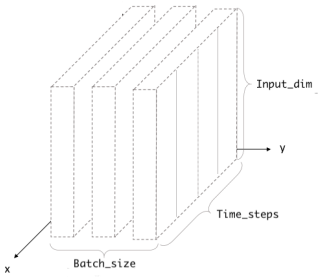
Figure 2: Model architecture

2.4 training procedure

For simplicity and lack of training resources we only consider a site (location) and trained a shallow model with 150 training epochs. The modelling goal is to generate forecasts for the next 24 hours at each respective hour of the day, therefore the model input was constructed to have a 24 hours window, see figure 3. That is, we use the previous 24 hours data to predict the next hour.



(a) Input window

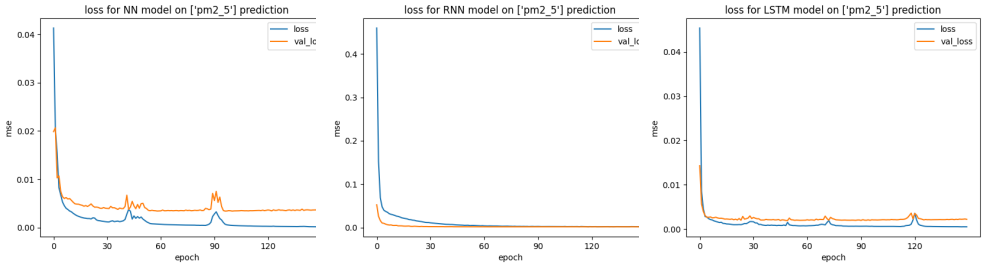


(b) input shape

Figure 3: Model input

3 Result and Analysis

Figure 4 shows the mean square error plots of each model after 150 epochs. Though the performance of the models are quite similar, the LSTM model came out to be the best models of all. LSTM are suitable for timeseries data as they remember relevant information from the input sequence. RNN are also good in this regards but become less performant when the shift between two consecutive inputs become wide.



(a) Neural Network (b) RNN (c) LSTM

Figure 4: Model mean square error

The models trained are stored for future prediction.

4 Conclusion/Extension

Though the goal here is to explore few machine learning analysis procedure for timeseries data we can, with time and resources, extend the analysis further. One of those things to try, apart from training deep neural networks, is to use extended classical models like ARIMA and SARIMA. Esemble modelling can also be explore, this combines different algorithms and aggregate the model predictions to result in one final prediction. Cross validation, which was not considered, is a good way to find best model parameters.

The github link for the analysis is accessible here `https://github.com/uman95/AirQo_assignment`.