**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**GRADUATION THESIS**

**Analysis and prediction of air quality data**

**NGUYEN DUC HIEU**

hieu.nd158130@sis.hust.edu.vn

**Information and Communications Technology**

|  |  |
| --- | --- |
| **Advisor/Instructor:** | Ph.D. Nguyen Ba Ngoc  Advisor’s signature |
| **Department:** | Information Systems |
| **School:** | Information and Communications Technology |

**HANOI, 06/2020**

**GRADUATE TOPIC**

Air pollution in big cities is an urgent problem. The problem is to integrate pollution-related data sources such as air monitoring systems, weather data sources, vehicles... to get a more comprehensive picture of pollution. From there, apply predictive models to predict air pollution in the future.

Advisor/Instructor

Sign and write full name

**Acknowledgements**

First of all, I would like to thanks Ph.D. Nguyen Ba Ngoc - School of Information and Communication Technology - Hanoi University of Science and Technology, for providing guidance and feedback throughout my dissertation.

Thanks also to other teachers at School of Information and Communication Technology and my colleagues for helping me complete the graduation thesis.

**Abstract**

Over the last few years, tackling air pollution is an urgent problem in many big cities like Hanoi. Much research is being conducted to evaluate the impact of air pollution on public health. Besides that, deterministic models of air pollutant behavior are also generated, however, these are both complex and often inaccurate. The main contents of this thesis are as follows:

## Objective: This project is designed to analyze the air quality index (AQI) dataset and use some methods to predict these values. After that, a number of experiments will be conducted. From the results received, I drawn some evaluation and proposed some further research directions.

## Main tasks:

## Introduce the forecasting task, get an overview of the air quality problem, the main air pollutants, including fine particulate matter (PM2.5), inhalable particle (PM10), ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2), and carbon monoxide (CO) and its effect on human health. Besides, have some basic knowledge about the Air Quality Index (AQI) and lastly the effect of weather to air quality.

* Research some theoretical methods necessary to perform data preparation and predict future values.

## Prepare the historical dataset for experiment and preprocessing the input data. From there, the data of AQI and weather values of the previous day are fitted to two models: Vector Autoregression and Long Short-Term Memory to predict the future AQI. Finally, a number of evaluations are made to extract some comparisons and conclusions.

Student

Sign and write full name

**TABLE OF CONTENTS**

[CHAPTER 1. AIR QUALITY PREDICTION PROBLEM 1](#_Toc44318725)

[1.1 Introduction to forecasting task 1](#_Toc44318726)

[1.2 An overview of air quality 2](#_Toc44318727)

[1.3 Air pollutants and their effects 2](#_Toc44318728)

[1.3.1 Particulate matter (PM10 and PM2.5) 3](#_Toc44318729)

[1.3.2 Ozone (O3) 4](#_Toc44318730)

[1.3.3 Oxides of nitrogen (NOx) 5](#_Toc44318731)

[1.3.4 Sulphur dioxide (SO2) 5](#_Toc44318732)

[1.3.5 Carbon monoxide (CO) 5](#_Toc44318733)

[1.4 Air Quality Index (AQI) 6](#_Toc44318734)

[1.4.1 Definition of AQI 6](#_Toc44318735)

[1.4.2 Converting concentration to AQI 7](#_Toc44318736)

[1.5 The influence of local meteorology on air quality 8](#_Toc44318739)

[1.5.1 Temperature 8](#_Toc44318740)

[1.5.2 Humidity 9](#_Toc44318741)

[1.5.3 Wind speed 9](#_Toc44318742)

[1.5.4 Pressure 9](#_Toc44318743)

[1.6 Some relevant studies on air quality forecasting 9](#_Toc44318744)

[1.7 Conclusion chapter 10](#_Toc44318745)

[CHAPTER 2. THEORETICAL METHODOLOGIES 11](#_Toc44318747)

[2.1 Data preprocessing 11](#_Toc44318748)

[2.1.1 Handle missing data 11](#_Toc44318749)

[2.1.2 Data Normalization 13](#_Toc44318750)

[2.2 Stationarity of time series 14](#_Toc44318751)

[2.3 Causality test 15](#_Toc44318752)

[2.4 Vector Autoregression (VAR) 16](#_Toc44318753)

[2.5 Deep learning and Long Short-Term Memory (LSTM) 17](#_Toc44318754)

[2.5.1 Overview of Neural Networks and Recurrent Neural Network 17](#_Toc44318755)

[2.5.2 Long Short-Term Memory (LSTM) 21](#_Toc44318756)

[2.6 Conclusion chapter 22](#_Toc44318757)

[CHAPTER 3. EXPERIMENTAL IMPLEMENTATION AND EVALUATION 23](#_Toc44318758)

[3.1 Data preparation 23](#_Toc44318759)

[3.2 Data preprocessing 24](#_Toc44318760)

[3.3 Testing stationarity 26](#_Toc44318761)

[3.4 Testing Granger’s Causality 27](#_Toc44318762)

[3.5 Vector Autoregression (VAR) 27](#_Toc44318763)

[3.6 Long Short-Term Memory 32](#_Toc44318764)

[3.7 Result summary 38](#_Toc44318765)

[CHAPTER 4. CONCLUSION 39](#_Toc44318766)

[4.1 Conclusion 39](#_Toc44318767)

[4.2 Direction for further research 39](#_Toc44318772)

[REFERENCES 40](#_Toc44318773)

**LIST OF FIGURES**

[Figure 1.1: Example of real time series (blue) and predicted series (red) 1](#_Toc44293889)

[Figure 1.2: Compares the sizes of PM10 and PM2.5 with other things 3](#_Toc44293890)

[Figure 1.3: The influance of different types of particulate matter 4](#_Toc44293891)

[Figure 1.4: Good vs Bad Ozone 4](#_Toc44293892)

[Figure 2.1: Some methods for handling missing value 12](#_Toc44293893)

[Figure 2.2: Architecture of a neural network 18](#_Toc44293894)

[Figure 2.3: Sigmoid function and its derivative 19](#_Toc44293895)

[Figure 2.4: A recurrent neuron unrolled through time 20](#_Toc44293896)

[Figure 2.5: LSTM cells and internal architecture 21](#_Toc44293897)

[Figure 3.1: AQI of PM2.5 from aqicn.org/data-platform/register/ 23](#_Toc44293898)

[Figure 3.2: Top five rows in the dataset 23](#_Toc44293899)

[Figure 3.3: AQI time series of 6 air pollutants 24](#_Toc44293900)

[Figure 3.4: % of missing values in 6 pollutants 25](#_Toc44293901)

[Figure 3.5: Imputed AQI time series of 6 air pollutants 25](#_Toc44293902)

[Figure 3.6: Summary of each pollutants 26](#_Toc44293903)

[Figure 3.7: A histogram for each pollutants 26](#_Toc44293904)

[Figure 3.8: Sub-data of PM2.5 27](#_Toc44293905)

[Figure 3.9: AIC of differents lag order 28](#_Toc44293906)

[Figure 3.10: MAE loss of VAR model 28](#_Toc44293907)

[Figure 3.11: Actual vs predicted PM2.5 of the next 1 day (VAR) 29](#_Toc44293908)

[Figure 3.12: Actual vs predicted PM10 of the next 1 day (VAR) 29](#_Toc44293909)

[Figure 3.13: Actual vs predicted O3 of the next 1 day (VAR) 29](#_Toc44293910)

[Figure 3.14: Actual vs predicted NO2 of the next 1 day (VAR) 30](#_Toc44293911)

[Figure 3.15: Actual vs predicted SO2 of the next 1 day (VAR) 30](#_Toc44293912)

[Figure 3.16: Actual vs predicted CO of the next 1 day (VAR) 30](#_Toc44293913)

[Figure 3.17: Actual vs predicted PM2.5 of the next 3 days (VAR) 31](#_Toc44293914)

[Figure 3.18: Actual vs predicted PM10 of the next 3 days (VAR) 31](#_Toc44293915)

[Figure 3.19: Actual vs predicted O3 of the next 3 days (VAR) 31](#_Toc44293916)

[Figure 3.20: Actual vs predicted NO2 of the next 3 days (VAR) 31](#_Toc44293917)

[Figure 3.21: Actual vs predicted SO2 of the next 3 days (VAR) 32](#_Toc44293918)

[Figure 3.22: Actual vs predicted CO of the next 3 days (VAR) 32](#_Toc44293919)

[Figure 3.23: Supervised data for LSTM 33](#_Toc44293920)

[Figure 3.24: LSTM model architecture 33](#_Toc44293921)

[Figure 3.25: Loss during traning of 6 time series 34](#_Toc44293922)

[Figure 3.26: Actual vs predicted PM2.5 of the next 1 day (LSTM) 34](#_Toc44293923)

[Figure 3.27: Actual vs predicted PM10 of the next 1 day (LSTM) 35](#_Toc44293924)

[Figure 3.28: Actual vs predicted O3 of the next 1 day (LSTM) 35](#_Toc44293925)

[Figure 3.29: Actual vs predicted NO2 of the next 1 day (LSTM) 35](#_Toc44293926)

[Figure 3.30: Actual vs predicted SO2 of the next 1 day (LSTM) 35](#_Toc44293927)

[Figure 3.31: Actual vs predicted CO of the next 1 day (LSTM) 36](#_Toc44293928)

[Figure 3.32: Loss during traning of 6 time series 36](#_Toc44293929)

[Figure 3.33: Actual vs predicted PM2.5 of the next 3 days (LSTM) 36](#_Toc44293930)

[Figure 3.34: Actual vs predicted PM10 of the next 3 days (LSTM) 37](#_Toc44293931)

[Figure 3.35: Actual vs predicted O3 of the next 3 days (LSTM) 37](#_Toc44293932)

[Figure 3.36: Actual vs predicted NO2 of the next 3 days (LSTM) 37](#_Toc44293933)

[Figure 3.37: Actual vs predicted SO2 of the next 3 days (LSTM) 37](#_Toc44293934)

[Figure 3.38: Actual vs predicted CO of the next 3 days (LSTM) 38](#_Toc44293935)

**LIST OF TABLES**

[Table 1.1: AQI scale table 7](#_Toc44293936)

[Table 1.2: EPA’s table of breakpoints 8](#_Toc44293937)

[Table 3.1: Python library packages 24](#_Toc44293938)

[Table 3.2: ADFuller Test result 26](#_Toc44293939)

[Table 3.3: Granger’s Causality Test result 27](#_Toc44293940)

[Table 3.4: Supervised form of data to predict 1 day-ahead AQI 33](#_Toc44293941)

[Table 3.5: Supervised form of data to predict 3 day-ahead AQI 33](#_Toc44293942)

[Table 3.6: MAE loss of the models when comparing actual and predicted values 38](#_Toc44293943)

# AIR QUALITY PREDICTION PROBLEM

## Introduction to forecasting task

Forecasting is a common statistical task in science and business, where it making predictions about the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts. In this thesis, we will focus on studying the time series prediction problem. A time series is a series of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. When forecasting time series data, the aim is to estimate how the sequence of observations will continue into the future.

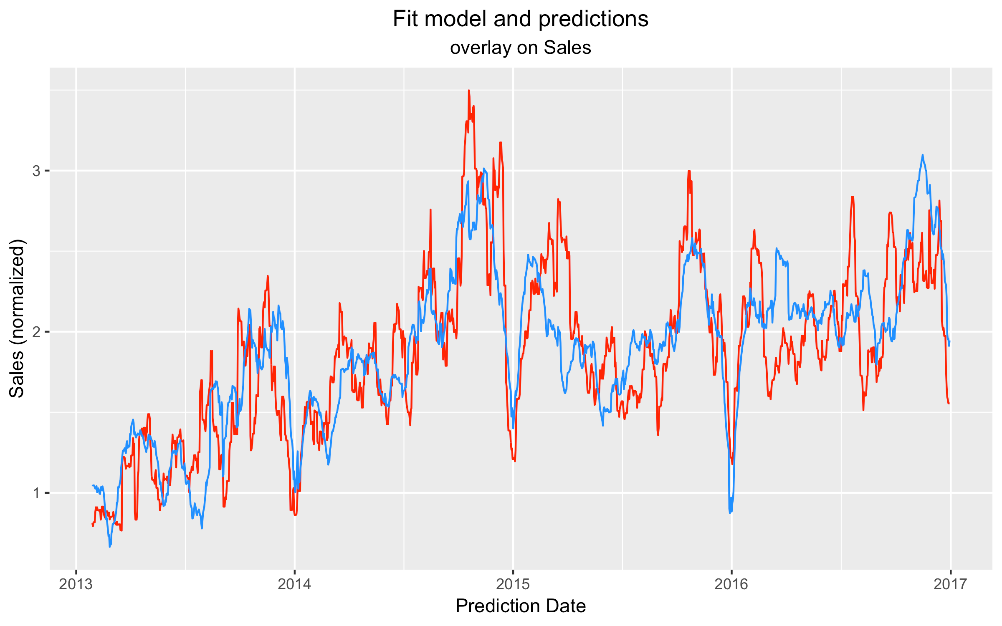


Figure .: Example of real time series (blue) and predicted series (red)

A forecasting task usually involves five basic steps:

* Problem definition: Often this is the most difficult part of forecasting. Defining the problem carefully requires an understanding of the way the forecasts will be used, who requires the forecasts, and how the forecasting function fits within the organisation requiring the forecasts.
* Gathering information: There are always at least two kinds of information required: (a) statistical data, and (b) the accumulated expertise of the people who collect the data and use the forecasts. Often, it will be difficult to obtain enough historical data to be able to fit a good statistical model.
* Preliminary analysis: Always start by graphing the data. Are there consistent patterns? Is there a significant trend? Is seasonality important? How strong are the relationships among the variables available for analysis?
* Choosing and fitting models: The best model to use depends on the availability of historical data, the strength of relationships between the forecast variable and any explanatory variables, and the way in which the forecasts are to be used. It is common to compare two or three potential models.
* Using and evaluating a forecasting model: Once a model has been selected and its parameters estimated, the model is used to make forecasts. The performance of the model can only be properly evaluated after the data for the forecast period have become available.

## An overview of air quality

Air quality has a huge impact on the quality of living, the well-being of the population as well as the image of the city. Air pollution exists around us and is very hard to avoid. Visible concentrations of smog in the air cannot reflect the freshness of the environment. In the past two years, from February to mid-April (the time the weather in Northern Vietnam often have heavy fog), information about air quality index (AQI) in Hanoi was always stirring up the community because of the pollution warning affecting people's health. Especially, at 14:00 on February 21, Air Visual, which applied the US AQI calculation, rank Hanoi as the world's most polluted air city out of 97 cities with air quality monitoring. The AQI value is 314, means that air pollution in Hanoi is at a dangerous level. This is the highest air quality warning threshold, harmful to the health of everyone. Hanoi was the only city in the world on that day where air quality was warned at a hazardous level.

In order to explain this problem, the reason was partly due to the weather with the thick fog, but most of the time due to the high traffic volume in the city center, the construction works were rushing up the schedule and other manufacturing operations also increased more, so it directly affected the quality of air. In Hanoi, there were some other specific causes, such as burning rice straw. This is a seasonal pollutant source but it is very harmful. Besides, Hanoi currently has more than 60,000 households using honeycomb charcoal stove, which is a very big number, along with the burning hazardous waste in suburban areas, the air pollution index in Hanoi in these months increased significantly.

So we see the importance of predicting air quality to protect human health. But first, we need to know what air quality is about. Let's take a closer look at some primary types of pollutants and their effects on people’s health.

## Air pollutants and their effects

According to WHO, 5 main substances causing air pollution that affect human health include:

* Particulate matter (PM10 and PM2.5)
* Ozone(O3)
* Oxides of nitrogen (NOx)
* Sulphur dioxide (SO2)
* Carbon monoxide (CO)

Pollutants can be either from natural sources or discharged in the atmosphere by human activities. Natural sources include emissions from plants, from the biomass of the ocean, volcanic gas and the re-suspension of dust in arid areas such as deserts. Man-made sources include combustion engines (both diesel and petrol), household and industry solidfuel combustion for energy production (coal, lignite, heavy oil and biomass), other industrial activities (building, mining, manufacture of cement, smelting), agriculture, with the use of entrants, and the erosion of roads by vehicles and abrasion of brakes and tyres…

### Particulate matter (PM10 and PM2.5)

Particulate matter (PM) consist of invisible solid and liquid particles with diameters of either from 2.5 μm to 10μm (PM10), or less than 2.5μm (PM2.5). They affect human more than any other pollutant, and can penetrate into the respiratory tract. PM2.5, being even smaller, can reach the deepest areas of the breathing apparatus, such as the pulmonary alveoli. In recent years, new types of PM has appeared: PM1.0 (under 1µm) and nano PM0.1 (below 0.1 µm), especially on low temperature days or dry air. The chemical compounds of PM include sulfates, nitrates, ammonium and other inorganic ions such as sodium, potassium, calcium or magnesium, metals such as cadmium, copper, nickel and zinc and biological components such as allergens or microbes.

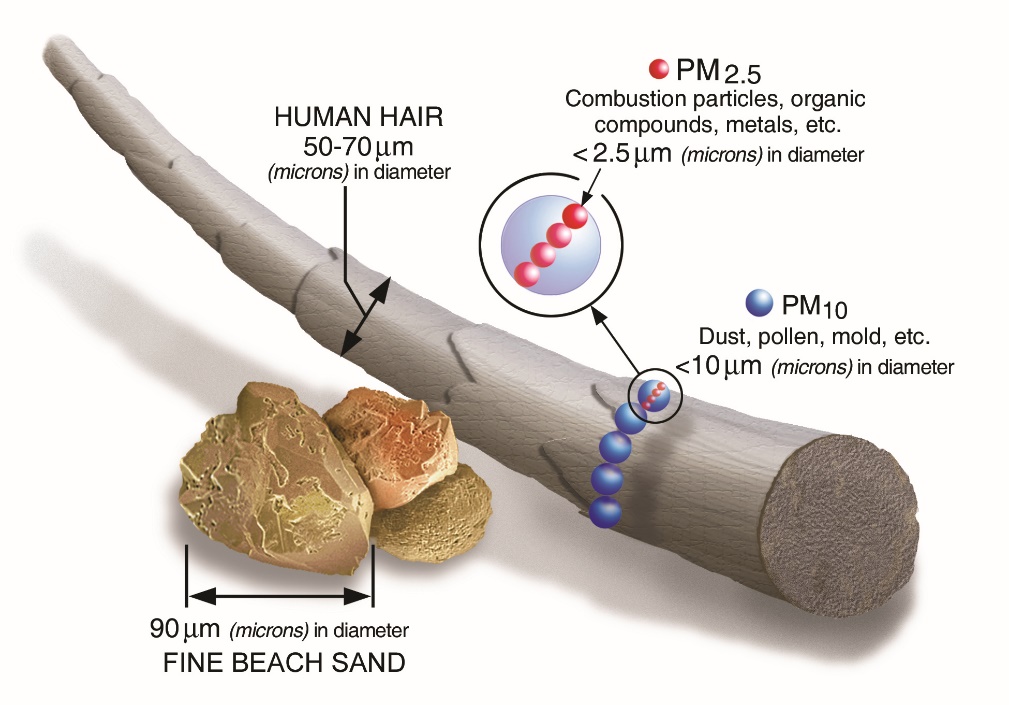


Figure .: Compares the sizes of PM10 and PM2.5 with other things

PM can be generated by industry, transport and agriculture, and due to their light

weight, can also be carried on air currents from one country to another. WHO recommend maintaining PM concentrations below the following levels [1]:

* PM10: 20 µg/m3 annual mean; 50 µg/m3 24-hour mean
* PM2.5: 10 µg/m3 annual mean; 25 µg/m3 24-hour mean

The International Agency for Research on Cancer (IARC) has demonstrated an increased risk of lung cancer with increased levels of exposure to PM and air pollution. PM2.5 and PM10 enter the respiratory tract when people breathe, but the level of penetration differs according to the size of their particles. While PM10 enters the body through the air passages and accumulates in the lungs, PM2.5 is especially dangerous because they are so small that they can slip into the alveolus, pulmonary veins and invade the circulatory system.

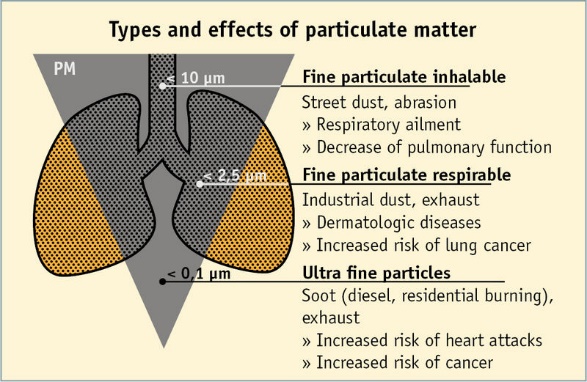


Figure .: The influance of different types of particulate matter

Long-term accumulation of PM10 and PM2.5 will increase the risk of disease in the respiratory system by reducing lung function, causing chronic bronchitis, asthma and lung cancer; damage cardiovascular system, circulatory system such as blood poisoning, blood clotting and human reproductive system. The epidemiologic impact of pollution on the population also apply to infants. Increased concentration of PM2.5 during pregnancy was associated with an increased risk of low birth weight, and babies are more likely to suffer from nervous breakdown and autism.

### Ozone (O3)

Ozone is a colorless gas found in the air we breathe. Ozone is good or bad, depending where it occurs. Good ozone is known as the high altitude shield of the Earth, where it protects the atmosphere against the harmful ultraviolet radiation emitted by the sun. However, bad ozone is a pollutant at lower altitudes resulting from a reaction between nitric oxides and organic volatile compounds (as hydrocarbons present in petrol, are released from vehicle exhaust and factories). This photochemical process can only occur under the radiation of the sun, which explains the summer-seasonality of ozone pollution events.

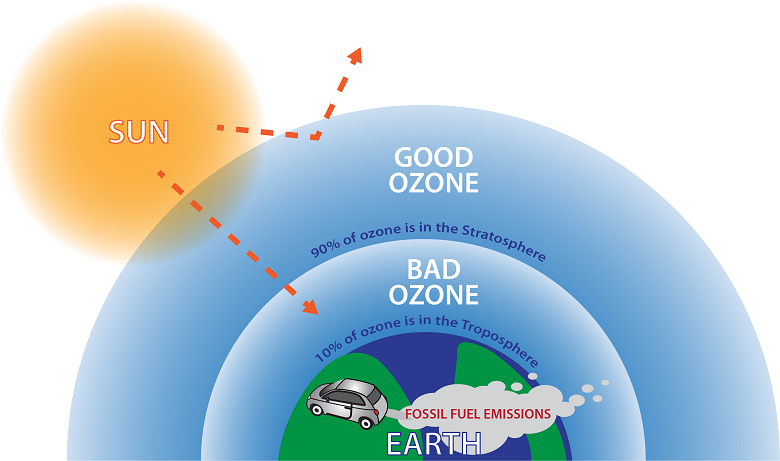


Figure .: Good vs Bad Ozone

When inhaled, ozone can damage the lungs. Relatively low amounts can cause chest pain, coughing, shortness of breath and throat irritation. Ozone may also worsen chronic respiratory diseases such as asthma and compromise the ability of the body to fight respiratory infections. Children are at higher risk from ozone exposure because they often play outdoors in summer when ozone levels are higher and their lungs are still developing. Repeated short-term ozone damage to children’s lungs may lead to reduced lung function in adulthood, which cause permanent lung damage. The recommended ozone concentration limit is [1]:

* O3: 100 µg/m3 8-hour mean

### Oxides of nitrogen (NOx)

Nitrogen oxides include nitric oxide (NO) and nitrogen dioxide (NO2), which are gases emitted from the burning of coal, oil, diesel fuel, and natural gas, especially from electric power plants. They are also emitted by cigarettes, gas stoves, kerosene heaters, wood burning, and silos that contain silage. In cities, most of the nitrogen oxides come from motor vehicle exhaust.

Nitric oxide (NO) is an important molecule in human cells, but has a limited toxicity in the concentrations at which it is found in the atmosphere. However, breathing nitrogen dioxide (NO2) can decrease lung function and increase the risk of respiratory symptoms like coughing and choking, nausea, headache and difficulty breathing… The concentration threshold that is safe for human health according to WHO guidelines is [1]:

* NO2: 40 µg/m3 annual mean; 200 µg/m3 1-hour mean

### Sulphur dioxide (SO2)

Sulphur dioxide (SO2) is a corrosive gas produced by the consumption of fuel containing sulphur, such as coal and oil. It can also be discharged in the atmosphere through natural processes, such as organic decomposition or volcanic eruptions. Sulphur dioxide irritates the skin and mucous membranes (eyes, nose, throat and lungs), and can affect the respiratory system. It causes coughing, mucus secretion and aggravates conditions such as asthma and chronic bronchitis.

Sulphur dioxide reacts with water in the air to form sulphuric acid, an important component of acid rain. Acid rain can cause deforestation, acidify waterways to the detriment of aquatic life and corrode building materials and paints. The WHO guidelines value of sulphur dioxide is [1]:

* SO2: 20 µg/m3 24-hour mean; 500 µg/m3 10-minute mean

### Carbon monoxide (CO)

Carbon monoxide is a colourless gas formed when substances containing carbon (such as petrol, gas, coal and wood) are burned with an insufficient supply of air. Motor vehicles are the main source of carbon monoxide pollution in urban areas. The emissions contain carbon monoxide from the incomplete burning of fuel in the engine.

Carbon monoxide has serious health impacts on humans and animals. CO in the air can be breathed in and absorbed easily through the lungs. Carbon monoxide combine with haemoglobin in red blood cells better than oxygen, therefore reduces the oxygen-carrying capacity of the red blood cells and decreases the supply of oxygen to tissues and organs, especially the heart and brain. For people with cardiovascular disease, this can be a serious problem. The most common symptoms of CO poisoning are headache, dizziness, weakness, upset stomach, vomiting, chest pain, and confusion. CO symptoms are often described as “flu-like.”

If you breathe in a lot of CO it can make you pass out or kill you. In Vietnam there are some tragic accident caused by CO poisoning in the freezing days when burning firewood to heat the house. Because in a closed room, the outside air cannot circulate, so the combustion process will not be provided with enough O2, resulting in the amount of CO in the combustion process increases gradually but cannot escape, it will accumulate near the floor (where we sleep), and silently kill us. What make CO even more dangerous is that CO is a colorless, odorless and non-irritating gas that makes it difficult to recognize the presence of CO in the air, especially when we are asleep. People can die from CO poisoning before they have any symptoms. The concentrations should be below the following levels:

* CO: 10 mg/m3 8-hour mean

## Air Quality Index (AQI)

### Definition of AQI

Since there are already recommended threshold values proposed by WHO, we should have a scale to quickly determine air pollution levels more intuitively without knowing the exact concentration. The Air Quality Index, or AQI, is a index a system used to warn the public when air pollution is dangerous and forecast their future values. It tracks many air pollutants, from many different sources (tiny particles from ash, power plants and factories, vehicle exhaust, soil dust, pollen, and other pollutants). The AQI for each city or region will be calculated based on the concentration of pollutants from different monitoring stations over a specified averaging period, and health effects corresponding to a given level are established by epidemiological research. Keeping track of the current air quality information can help you take steps to protect yourself, children, and others from unhealthy levels of air pollution. In many countries across the world, air pollution levels are measured daily and ranked on a scale of 0 for perfect air all the way up to 500 for air pollution levels that pose an immediate danger to the public. Think of the AQI as a yardstick that runs from 0 to 500. The higher the AQI value, the greater the level of air pollution and the greater the health concern.

The AQI is divided into six categories. Each category corresponds to a different level of health concern. Each category also has a specific color. The color makes it easy for people to quickly determine whether air quality is reaching unhealthy levels in their communities.

Table .: AQI scale table

|  |  |  |  |
| --- | --- | --- | --- |
| Daily AQI Color | Levels of Concern | Values of Index | Description of Air Quality |
| Green | Good | 0 to 50 | Air quality is satisfactory, and air pollution poses little or no risk. |
| Yellow | Moderate | 51 to 100 | Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution. |
| Orange | Unhealthy for Sensitive Groups | 101 to 150 | Members of sensitive groups may experience health effects. The general public is less likely to be affected. |
| Red | Unhealthy | 151 to 200 | Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects. |
| Purple | Very Unhealthy | 201 to 300 | Health alert: The risk of health effects is increased for everyone. |
| Maroon | Hazardous | 301 and higher | Health warning of emergency conditions: everyone is more likely to be affected. |

United States Environmental Protection Agency (EPA) establishes an AQI for five major air pollutants we discuss earlier:

* Particulate matter (PM10 and PM2.5)
* Ozone(O3)
* Oxides of nitrogen (NOx)
* Sulphur dioxide (SO2)
* Carbon monoxide (CO

Daily overall AQI is considered to be the highest AQI value among these pollutants:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

### Converting concentration to AQI

The AQI is a piecewise linear function of the pollutant concentration. At the boundary between AQI categories, there is a discontinuous jump of one AQI unit. To convert from concentration to AQI this equation is used:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

## Where:

* = Air Quality Index
* = the pollutant concentration
* = the concentration breakpoint that is ≤
* = the concentration breakpoint that is ≥
* = the index breakpoint corresponding to
* = the index breakpoint corresponding to

## EPA's table of breakpoints is:

Table .: EPA’s table of breakpoints

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0-50 | 51-100 | 101-150 | 151-200 | 201-300 | 301-400 | 401-500 |
| PM2.5  (μg/m3) | 0.0-12.0 (24-hr) | 12.1-35.4 (24-hr) | 35.5-55.4 (24-hr) | 55.5-150.4 (24-hr) | 150.5-250.4 (24-hr) | 250.5-350.4 (24-hr) | 350.5-500.4 (24-hr) |
| PM10  (μg/m3) | 0-54 (24-hr) | 55-154 (24-hr) | 155-254 (24-hr) | 255-354 (24-hr) | 355-424 (24-hr) | 425-504 (24-hr) | 505-604 (24-hr) |
| O3  (ppb) | 0-54  (8-hr) | 55-70  (8-hr) | 71-85  (8-hr) or 125-164  (1-hr) | 86-105  (8-hr) or 165-204  (1-hr) | 106-200  (8-hr) or 205-404  (1-hr) | 405-504  (1-hr) | 505-604  (1-hr) |
| NO2  (ppb) | 0-53  (1-hr) | 54-100  (1-hr) | 101-360 (1-hr) | 361-649 (1-hr) | 650-1249 (1-hr) | 1250-1649 (1-hr) | 1650-2049 (1-hr) |
| SO2  (ppb) | 0-35  (1-hr) | 36-75  (1-hr) | 76-185 (1-hr) | 186-304 (1-hr) | 305-604 (24-hr) | 605-804 (24-hr) | 805-1004 (24-hr) |
| CO  (ppm) | 0.0-4.4 (8-hr) | 4.5-9.4 (8-hr) | 9.5-12.4 (8-hr) | 12.5-15.4 (8-hr) | 15.5-30.4 (8-hr) | 30.5-40.4 (8-hr) | 40.5-50.4 (8-hr) |

For example, suppose a monitor records a 24-hour average particulate matter 2.5 (PM2.5) concentration of 10.0 micrograms per cubic meter (μg/m3). Based on the equation above and the EPA’s table of breakpoints, the calculation of the AQI will be:

## The influence of local meteorology on air quality

The weather can have a significant impact on air quality since different aspects of the weather affect the amounts pollutants that are present in a specific area. Let's take a look at some weather factors that are likely to affect the air quality.

### Temperature

Air temperature affects the movement of air, and thus the movement of air pollution. The warmer, lighter air at the surface rises and the cooler, heavier air in the upper troposphere sinks. Cold temperatures in the winter can lead to worsened air quality. Particulate matter (PM10 and PM2.5) and carbon monoxide (CO) concentrations from wood burning increases during the cold winter months because everyone stay indoor. Besides, cold temperatures can trap pollutants near the ground through a process called “temperature inversion.” This happens when a layer of warmer air sits above the colder air at the surface, acting like a lid that traps in pollution. This is how things like smog, smoke, and carbon monoxide (CO) can stay around for long periods of time, and they usually don’t get broken up until a weather event (wind, rain, snow) comes through the area.

Higher air temperatures can speed up chemical reactions in the air, such as the reactions that create harmful ozone (O3) is more efficiently in sunny day:

NOx + VOC + Heat&Sunlight -> O3

Heat waves often lead to poor air quality. The extreme heat can increase the amount of forest fires. Fires add CO and particle pollution to the atmosphere. Heatwaves often come with high pressure, that creates a layer of air above ground level. When this happens on cities, air pollutants remain captured, and pollutants density increases.

### Humidity

Humidity can help to decrease ozone (O3) pollution. High humidity creates clouds that block sunlight, causing O3 production to slow down for the day, moisture and water vapor destroys the ozone that has formed. High humidity may also be due to a rain, and that rain washes out some pollutants and particulate matter. It also be noted that absolute humidity is very strongly correlated with temperature, so it may be reflecting temperature effects.

### Wind speed

Wind carries air contaminants away from their source, causing them to disperse. When wind speeds are low, pollutants tend to pile up in calm conditions. The higher the wind speed, the more contaminants are dispersed and the lower their concentration. However, high wind can also generate dust – a problem in dry windy rural areas.

### Pressure

We can see from the section above, high pressure goes along with heatwaves creating stagnant air that causes pollutants such as vehicle and factory exhaust concentrate over an area. As for low pressure systems, the weather is often wet and windy, causing pollutants to be dispersed or washed out of the atmosphere by rain.

## Some relevant studies on air quality forecasting

Different methodologies have been applied to describe and forecast the dispersion of air pollutants, from the most traditional statistical approaches, such as ordinary least squares (OLS) and first-order autocorrelation (AR) model of Shi and Harrison (1997) [2], to the Autoregressive Integrated Moving Average (ARIMA) model adopted by Abhilash et al. (2007) [3]. Along with the great development of machine learning and deep learning in recent years, many other studies have used these techniques to predict time series concentrations. Bing-Chun Liu et al. [4] proposed new model of collaborative forecasting using Support Vector Regression (SVR) for Urban Air Quality Index (AQI) prediction in three cities of China. Artificial Neural Networks (ANN) have been shown to provide better predictive results than linear models such as multiple linear regression (MLR) and time-series models such as Autoregressive Integrated Moving Averages (ARIMA) (Victor R. Prybutok et al. (2000) [5], A. Vlachogianni et al. (2011) [6]). ANN models are often considered to be a good compromise between simplicity and efficiency, being able to model the effects of non-linearities. The only drawbacks of neural networks is their flexibility: there are many hyperparameters for tuning. Even in a simple network you can change the number of layers, the number of neurons per layer, the type of activation function…

For time series data, Recurrent Neural Networks (RNN) and Long Short-Term Memory in particular are well-suited. LSTMs have been used in many time series applications, including rhythm learning, speech recognition, sign language translation, and much more. As for air quality problem, Yi-Ting Tsai et al. [7] proposed an LSTM approach to effectively forecast the value of PM2.5 in Taiwan. Ricardo Navares and José L. Aznarte [8] use four differents LSTM architectures to predict some main air pollutants and airborne pollen concentrations of two genus in Madrid. These models gives a positive result and there are still room for improvement.

## Conclusion chapter

## In this chapter we have learned the basics of the forecasting task and some information related to air quality. A number of prediction models have been applied by the researchers to predict the AQI values. In the next chapter, we will talk about some theoretical methods used in this thesis to carry out this prediction problem.

# THEORETICAL METHODOLOGIES

## Data preprocessing

Assuming you already have a dataset, the first step you should take is preprocessing the data, which is a very important step. Data preprocessing will greatly affect the accuracy of the model, since mistakes, redundancies and missing values all compromise to produce biases and variances.

### Handle missing data

Most machine learning algorithms cannot work with missing values, so handle missing data is one of the most important thing to do in data preprocessing. In general, there are three types of missing data according to the mechanisms of missingness.

* Missing completely at random (MCAR): Missing completely at random means that the missing data is completely unrelated to the other information in the data. There is no relationship between the missingness of the data and any values, the data is missing has nothing to do neither with observed data nor with non-observed data, it’s just missing. There is no logic in it. For example, a recorded data is lost because of a problem with the system.
* Missing at random (MAR): Missing at random means that the reason the for missingness is not related to the specific missing data, but it is related to some of other features and observed data and we can have a prediction from that. For example, only younger people have missing values for salaries.
* Missing not at random (MNAR): Missing not at random means that missing data is related to the unobserved data, the data that we don’t have, the missingness is related to factors that we didn’t consider. For example, people with high salaries generally do not want to reveal their incomes in surveys.

In the first two cases, it is safe to remove the data with missing values depend on their occurrences, while the third case are problematic. The only way to obtain an unbiased estimate of the parameters in such a case is to model the missing data. Two main techniques for handling the missing data is either deletion (drop columns or rows with missing data) or imputation (fill in the missing value). Some possible methods is presented below:

Figure .: Some methods for handling missing value

Let’s take a brief look at the methods mentioned above:

* Listwise Deletion: if missing values in some entries in the dataset is missing completely at random, we can delete all data from that entries.
* Pairwise Deletion: is an alternative to listwise deletion to mitigate the loss of data. It’s only deletes cases when one of the variables in the particular model you are evaluating is missing. It assumes that the missing data are missing completely at random.
* Mean/Median/Mode Imputation: when data is missing at random, we can replace the missing values with the mean/median/mode of the non-missing values.
* Last Observation Carried Forward (LOCF) & Next Observation Carried Backward (NOCB): just like their name, LOCF uses last observed value to fill in all subsequent missing points, similarly, NOCB use next observed value to fill in all previously missing points.
* Linear Interpolation: is an imputation method that assumes a linear relationship between data points and uses non-missing values from adjacent data points to compute a value for a missing data point.
* K-Nearest Neighbors: finding the k-closest neighbours to the entries with missing data and then imputing them based on the non-missing values in the neighbourhood.
* Multiple Imputation: impute the missing entries of the incomplete data sets multiple times, then analyze each of the completed data sets and finally integrate the analysis results into a final result. This is the most complicated and sophisticated method.

In my particular case, the percentage of missing data is fairly large, so the deletion methods will not be suitable as it make the dataset much smaller and there is not enough data to fit the models. For this reason, i will choose the K-Nearest Neighbors (KNN) imputation, as it is widely used. It is a simple algorithm but not as simple as mean or median imputation, also it produces quite good results. In this method, k-neighbors are chosen based on some distance measure and their average is used as an imputation estimate. For example, a missing data point for PM10 AQI will be imputed by the average of k PM10 AQI values of k-most similar observations which have similar other AQI and weather values. When using KNN, you have to take two main parameters into consideration:

* The number of neighbors: Taking a low k will increase the influence of noise and the results are going to be biased. On the other hand, taking a high k will tend to blur local effects. If you have no idea what the correct value of k is, it is recommended to try an small odd k and increase k gradually.
* Distance metric: The commonly used distance metrics for continuous data are Euclidean, Manhattan and Cosine. Euclidean is a good distance measure to use if the input variables are similar in type (e.g. all measured widths and heights). Manhattan distance is a good measure to use if the input variables are not similar in type (such as age, height, etc…).

### Data Normalization

Other important preprocessing steps you need to apply to your data is data normalization or feature scaling. With few exceptions, Machine Learning algorithms don’t perform well when the input numerical attributes have very different scales. There are two common ways to get all attributes to have the same scale: min-max scaling and standardization:

* Min-max scaling: values are shifted and rescaled so that they end up ranging from 0 to 1. We do this by subtracting the min value and dividing by the max minus the min:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

* Standardization: first it subtracts the mean value (so standardized values always have a zero mean), and then it divides by the standard deviation so that the resulting distribution has unit variance. Unlike min-max scaling, standardization does not bound values to a specific range, which may be a problem for some algorithms. However, standardization is much less affected by outliers:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Where: = mean value; = standard deviation

Because neural networks expect an input value ranging from 0 to 1, I choose min-max scaling for this AQI data.

## Stationarity of time series

A stationary time series is a time series whose statistical properties, such as mean and variance, do not change over time. Time series with trends, or with seasonality, are not stationary - the trend and seasonality will affect the value of the time series at different times [9]. A stationary time series tends to return to the mean and the fluctuations around the mean will be the same. Why does a time series have to be stationary? Like in the case of dependence and independence variables, the same goes for stationarity and non-stationarity. Two independence random variables (X1, X2) cannot give us any idea of pattern to follow up for prediction, but they can be dependent in various form, for example: X1 = a + bX2,X1 = X22… Stationarity is just that form/model out of all the dependency models we use to find the relation between the samples in a time series. With non-stationary time series, it can take any form, the mean, the variance will change through time, so it cannot be modeled. In conclusion, stationarity make us easy to analyze and model time series. There is one way to make a non-stationary time series stationary: differencing - compute the differences between consecutive observations. Differencing can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

To check if a time series is stationary or not, a test called unit root test is performed. It is a hypothesis test with null hypothesis stating that a unit root exists in time series sample, which means non-stationarity, and the alternative hypothesis is stationarity. A unit root (also called a unit root process or a difference stationary process) is a stochastic trend in a time series, sometimes called a “random walk”. A common unit root test I will use in this thesis is called Augmented Dickey–Fuller test. Specifically, the ADF test model has three main versions:

* Test for a unit root:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

* Test for a unit root with drift:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

* Test for a unit root with drift and deterministic time trend:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Where:

* : the observations in original series
* the differenced series
* : a constant
* : the coefficients
* : the lag order of the autoregressive process
* : white noise

Like other statistical hypothesis tests, a critical value is compared to the test statistic to determine whether to reject the null hypothesis. If the absolute value of your test statistic is greater than the critical value, you can declare statistical significance and reject the null hypothesis. Each version of the test has its own critical value which depends on the size of the sample. Our testing hypotheses are:

* Null hypothesis H0: = 0 (which means the series is non-stationary)
* Alternative hypothesis Ha: < 0 (which means the series is stationary)

The intuition behind the test is as follows. If the series is stationary, then it has a tendency to return to a constant (or deterministically trending) mean. Therefore, large values will tend to be followed by smaller values (negative changes), and small values by larger values (positive changes). Accordingly, the level of the series will be a significant predictor of next period's change, and will have a negative coefficient. If, on the other hand, the series is non-stationary, then positive changes and negative changes will occur with probabilities that do not depend on the current level of the series; like in a random walk, where you are now does not affect which way you will go next.

## Causality test

To check each of the weather data in the system influences the air quality or not, I will use Granger’s Causality Test. In other words, you can predict the AQI series with past values of itself along with other weather series in the system. Using this test, it’s possible to test this relationship before even building the model. According to Granger causality, variable that evolves over time “Granger-causes” another evolving variable if predictions of the value of based on its own past values and on the past values of are better than predictions of based only on own past values.

Granger’s Causality Test is also a hypothesis test. Let and be stationary time series. To test the null hypothesis that does not “Granger cause” , the two time series is fitted to the bivariate regressions with time lags as follows:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

The null hypothesis (H0) is that all the coefficients . In contrast, is called a “Granger cause” of if at least one of the coefficients for i = 1,…,p is significantly larger than zero (in absolute value), which is the alternative hypothesis (Ha) of the test. If the p-value obtained from the test is less than the significance level of 0.05, then, we can safely reject the null hypothesis.

## Vector Autoregression (VAR)

Vector Autoregression (VAR) is a stochastic process model used to capture the linear interdependencies among multivariate time series. It can be considered an extension of the Autoregressive (AR) model. One limitation of the Autoregressive (AR) model and Autoregressive Integrated Moving Averages (ARIMA) model is that they impose a unidirectional relationship — the forecast variable is influenced by the predictor variables, but not vice versa. Whereas, Vector Auto Regression (VAR) is bi-directional. In this model, all variables are treated symmetrically. They are all modelled as if they all influence each other equally [9]. The vector autoregression (VAR) model extends the idea of univariate autoregression to time series regressions, where the lagged values of all series appear as regressors. A typical AR(p) model equation is:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Where:

* : a constant
* ­, ­, …, ­: the coefficients
* : the lag order
* : white noise

In the VAR model, each variable is modeled as a linear combination of past values of itself and the past values of other variables in the system. Since you have multiple time series that influence each other, it is modeled as a system of equations with one equation per variable (time series). A p-th order VAR, denoted VAR(p), for n variables as follows:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

For example, a second order VAR(2) model with two variables ( and ) would look like this:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Just like other linear regression models, ordinary least squares estimation (OLS) is used to estimate the coefficients of the model. There are two things we has to choose when using a VAR to forecast, namely how many variables (denoted by n) and how many lags (denoted by p) should be included in the model. For example, for a VAR with n=5 variables and p=3 lags, there are 16 coefficients per equation, giving a total of 80 coefficients to be estimated. The more coefficients that need to be estimated, the larger the estimation error entering the forecast. In practice, it is usual to keep p small and include only variables that are correlated with each other, and therefore useful in forecasting each other. To select the lag order, Information criterion like Akaike information criterion (AIC), Bayesian information criterion (BIC), Hannan–Quinn information criterion (HQIC) or Final Prediction Error (FPE) are commonly used.

In this thesis, I will choose AIC to select the number of lags of VAR model. Akaike Information Criterion (AIC) is a mathematical method for evaluating how well a model fits the data it was generated from. AIC is used to compare different possible models and determine which one is the best fit for the data. A good model is the one that has minimum AIC among all the other models. To estimate the AIC, we use the following equation:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Where:

* P: the number of parameters learned by the model
* L: the maximized value of the likelihood function of the model

## Deep learning and Long Short-Term Memory (LSTM)

### Overview of Neural Networks and Recurrent Neural Networks (RNNs)

#### Artificial Neural Networks

Artificial neural networks are popular mathematical models applied in machine learning today. The architecture of a neuron network is described like a brain, consisting of a group of connected neurons (nodes), which process information by spreading information of computational processes in machine learning. They can be used to model, find complex relationships between input data and output data, or to find the patterns in data. Artificial neural networks usually include these components: network nodes (including weights), layers (a set of several nodes), activation functions, and lost functions. Besides that, depending on the architecture, different networks may have additional hyperparameters.

In a neural network, each node is related to each other, and impacts on data received from other nodes by parameters, these parameters will be updated in the learning process to obtain the solution model. The architecture of a neural networks is described as follows:

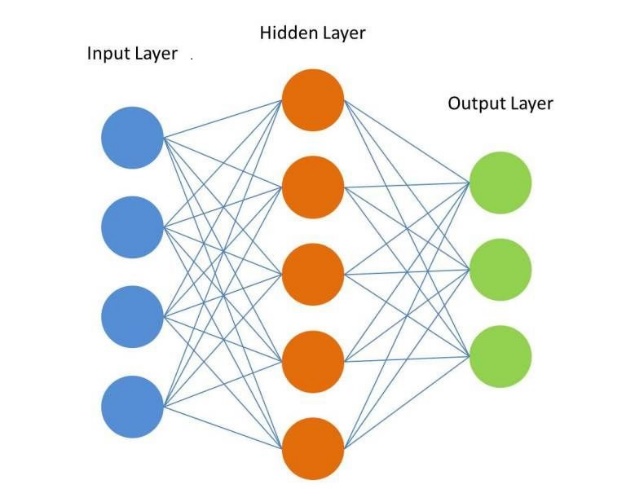


Figure .: Architecture of a neural network

The network layer will consist of many nodes processing data at the same level of abstraction. The input layer is the vector for encoding learning input data (the lowest level of abstraction). The ouput layer is a representation of the desired output of the problem. Today’s neural networks often have one or more layers between the input and output layers, called hidden layers. Every layer except the output layer includes a bias neuron and is often fully connected to the next layer.

To train a neural network, an algorithm called backpropagation algorithm is used. In short, it is Gradient Descent using an efficient technique for computing the gradients automatically in just two passes through the network (one forward, one backward), the backpropagation algorithm is able to compute the gradient of the network’s error with regard to every single model parameter. In other words, it can find out how each connection weight and each bias term should be tweaked in order to reduce the error. Once it has these gradients, it just performs a regular Gradient Descent step, and the whole process is repeated until the network converges to the solution.

Firstly, each input is passed to the network’s input layer, which sends it to the first hidden layer. The algorithm then computes the output of all the neurons in this layer (for every instance in the mini-batch). The result is passed on to the next layer, its output is computed and passed to the next layer, and so on until we get the output of the last layer, the output layer. This is called the feedforward pass:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Next, the algorithm measures the network’s output error (it uses a loss function that compares the predicted output and the actual output of the network). Then it computes how much each output connection contributed to the error. This is called backpropagation pass and is done analytically by applying the chain rule until the algorithm reaches the input layer:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Finally, the algorithm performs a Gradient Descent step to tweak all the connection weights in the network, using the error gradients it just computed.

In order for this algorithm to work properly, the step function f is a non-linear function like sigmoid function:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

or hyperbolic tangent function (tanh) and ReLU. Without this activation function, it can be seen that the output of each layer will be linearly dependent on the input, all layers will only receive values equal to the product of the weight matrix with the output of the previous layer, so the output of the whole network will also be linearly dependent on the input. One problem is that the sigmoid activation function (or tanh) saturates at 0 or 1 when inputs become large (negative or positive), with a derivative extremely close to 0:

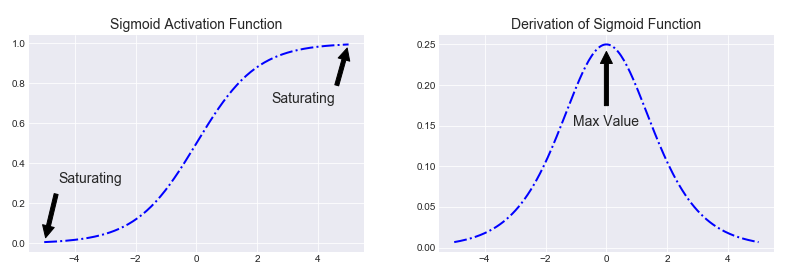


Figure .: Sigmoid function and its derivative

When the derivative is very small (closed to 0) and we do matrix multiplication, the gradient often get smaller and smaller as the algorithm progresses down to the lower layers. As a result, the Gradient Descent update leaves the lower layers’ connection weights unchanged, and training never converges to a good solution. We call this the vanishing gradients problem. To solve this problem, we usually use ReLU function instead of tanh or select an appropriate weight initialization.

#### Recurrent Neural Networks

In traditional neuron networks all inputs and outputs are independent of each other. That is, they are not create sequences. These models do not fit in many time series problems. For example, if you want to guess the next word that might appear in a sentence, you also need to know how the previous words appear. Recurrent Neural Networks (RNNs) address this issue. RNNs has the ability to remember previously calculated information. They are networks with loops in them, allowing information to persist.

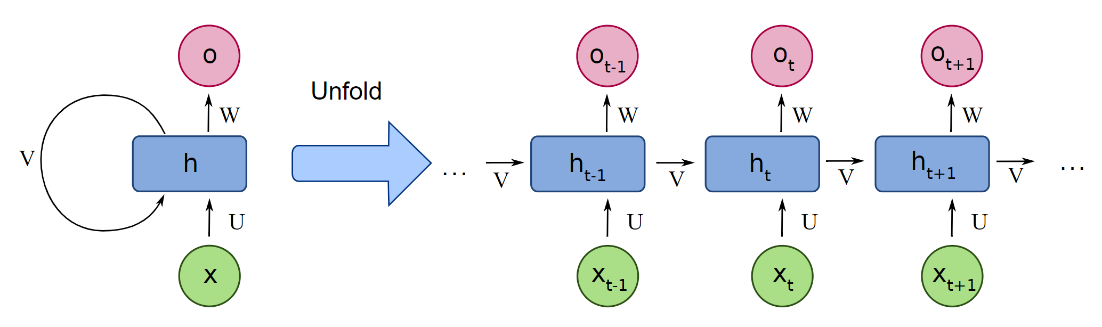


Figure .: A recurrent neuron unrolled through time

Let’s look at the simplest possible RNN, composed of one neuron receiving inputs, producing an output, and sending that output back to itself. At each time step t (also called a frame), this recurrent neuron receives the inputs as well as its own output from the previous time step to produce the output . Since there is no previous output at the first time step, it is generally set to 0. is the hidden state in t. It acted like the memory of the network and is calculated based on both the previous state and the input at that step:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

The function is usually a non-linear function like hyperbolic tangent function (tanh) or ReLU. Training RNN is similar to traditional neural networks, however the backpropagation algorithm must change a bit. The derivative at each output depends not only on the calculations at that step, but also on the previous steps, as parameters in the RNN are used for all steps in the network. This type of derivative calculation is called Backpropagation Through Time. With the loss function , to calculate the derivative, we use the chain rule. We see that the derivative for is simply matrix multiplication:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

However, with and , our calculation is not that simple:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Because , depends on , depends on and so on:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

This calculation is the same as the backpropagation algorithm in a traditional neural network. The difference is that we sum the derivatives of at each time step. Similar to traditional neural networks, vanishing gradients problem also happen with RNN and cause Short-Term memory. When traversing an RNN, the gradient is smaller at each time step. After a while, the RNN’s state contains no trace of the first inputs. To tackle this problem, Long Short-Term Memory networks (LSTM) have been introduced.

### Long Short-Term Memory (LSTM)

LSTMs are explicitly designed to avoid the long-term dependency problem. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTM also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four layer interacting in a very special way. Its architecture is shown as below:

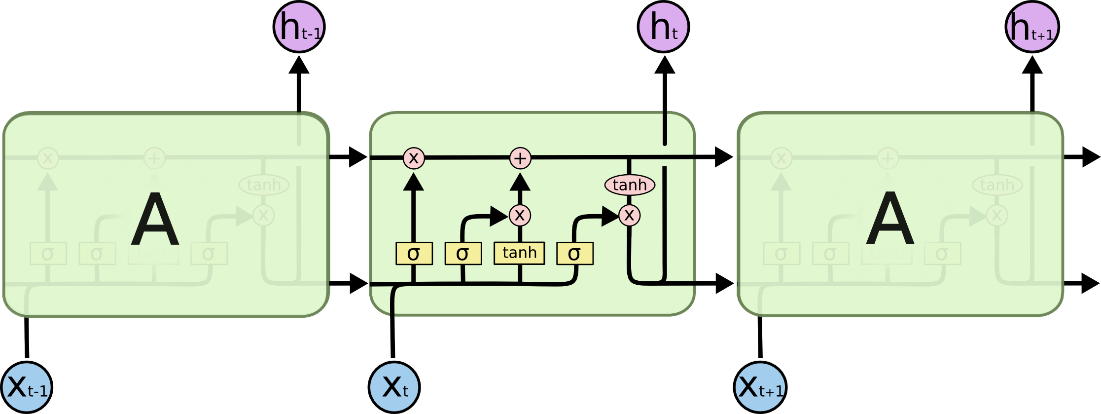


Figure .: LSTM cells and internal architecture

The core concept of LSTM are the cell state and it’s various gates. LSTM’s state is split into two vectors: and . You can think of as the short-term state and as the long-term state. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. So even information from the earlier time steps can make it’s way to later time steps, reducing the effects of short-term memory. As the cell state goes on its journey, information get’s added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training.

Let’s take a look at various gates of LSTM. The first gate is called the “forget gate.” This gate decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function. Values come out between 0 and 1 for each number in the cell state . The closer to 0 means to forget, and the closer to 1 means to keep.

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

To update the cell state, we have the input gate. First, we pass the previous hidden state and current input into a sigmoid function. That decides which values will be updated by transforming the values to be between 0 and 1, 0 means not important, and 1 means important. You also pass the hidden state and current input into the tanh function as a activation function. Then you multiply the tanh output with the sigmoid output . The sigmoid output will decide which information is important to keep from the tanh output.

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Now it’s time to update the old cell state into the new cell state . First, the old state gets pointwise multiplied by the forget vector , forgetting the things we decided to forget earlier. Then we add the result with the pointwise multiplication of the input gate with , which updates the cell state to new values that the neural network finds relevant. That gives us our new cell state.

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Finally we have the output gate. The output gate decides what the next hidden state should be. This output will be based on our cell state, but will be a filtered version. First, we pass the previous hidden state and the current input into a sigmoid function. Then, we put the cell state through tanh and multiply it by the output of the sigmoid gate to decide what information the hidden state should carry. The new hidden is then carried over to the next time step.

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

## Conclusion chapter

In this chapter we talk about a number of theoretical methods include including data preprocessing and the models models that will be used to predict the AQI values. In the next chapter, these methods will be implemented and experimented on a real dataset.

# EXPERIMENTAL IMPLEMENTATION AND EVALUATION

## Data preparation

The dataset I chose for this experiment is the historical dataset of AQI in Hanoi and can be downloaded from [aqicn.org/data-platform/register/](https://aqicn.org/data-platform/register/) [10] and were provided by the World Air Quality Index (WAQI) team and Vietnam Center For Environmental Monitoring Portal. The dataset consist of daily average AQI, not the raw concentrations, for the six individual air pollutants PM2.5, PM10, O3, NO2, SO2 and CO in Hanoi from 03-01-2016 to 31-05-2020. An intuitive view of the dataset as below:

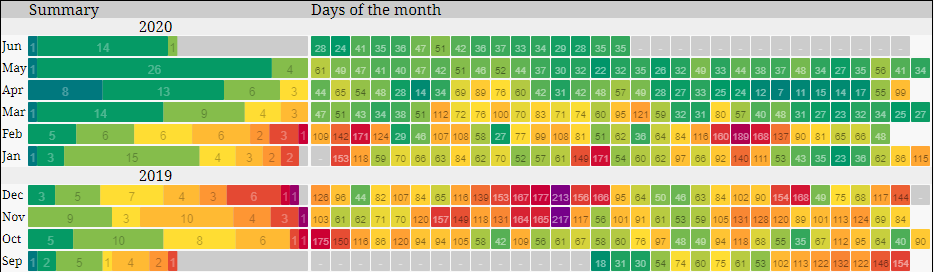


Figure .: AQI of PM2.5 from [aqicn.org/data-platform/register/](https://aqicn.org/data-platform/register/)

Weather observations from the nearest airport weather station was downloaded from a public dataset from Reliable Prognosis [11]. It includes the following data:

* Hourly air temperature *T* (degrees Celsius - °C) at 2 meters height above the earth’s surface.
* Hourly atmospheric pressure *P0* at weather station level (millimeters of mercury - mmHg)
* Hourly relative humidity *U* (%) at 2 meters height above the earth’s surface.
* Hourly mean wind speed *Ff* at a height of 10-12 meters above the earth’s surface over the 10-minute period immediately preceding the observation (meters per second – m/s)

Both datasets is in the form of csv files. The only problem is each observation of AQI is based on a day, while weather data is hourly, so I have manually taken some extra quick steps in MS Excel to get daily weather data. Finally, by merging the two datasets, we have the full dataset to experiment with. Let’s take a look at the top five rows of the dataset:

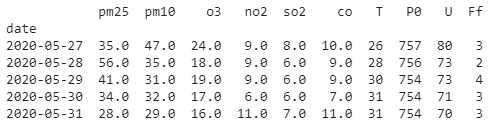


Figure .: Top five rows in the dataset

Next the dataset is split into training set, validation set and test set in order to train and evaluate the performance of the models. The test set will not be used during the training phase.

* Training set: data from January 2016 until December 2018
* Validation set: 12 months between January 2019 and December 2020
* Test set: 5 months of 2020

Before fitting the data to the chosen models, some pre-processing should be performed for time series data.

## Data preprocessing

The experimental implementation is written in the python programming language version 3.7 running on 64-bit windows 10 operating system. In particular, library packages are listed as below:

Table .: Python library packages

|  |  |
| --- | --- |
| Library package name | Description |
| pandas | Data manipulation and analysis |
| os | Interacting with the operating system |
| numpy | Scientific computing |
| matplotlib | Creating static, animated, and interactive visualizations |
| scikit-learn | Machine learning library |
| statsmodels | Providing statistical models |
| tensorflow | Machine learning and neural networks |

First we should have an intuitive view of the 6 air pollutants’s time series:

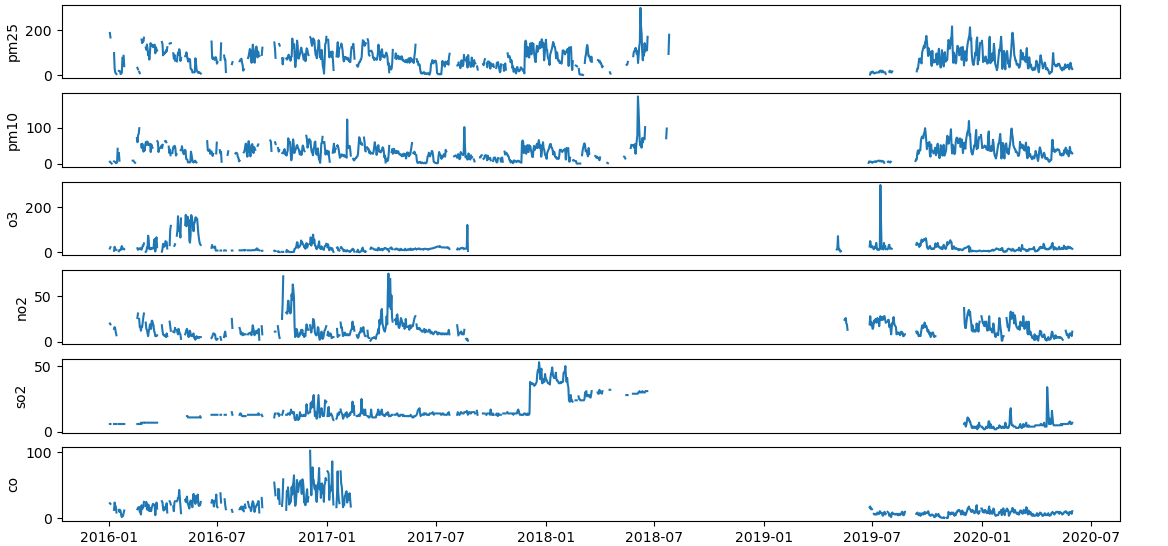


Figure .: AQI time series of 6 air pollutants

We can see that from mid 2017, early 2018 to early 2019, the amount of lost data was quite large. Percentage of missing values in 6 pollutants is shown in the bar chart below:

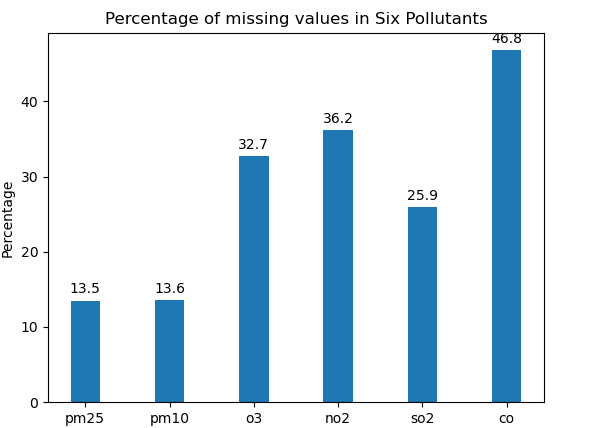


Figure .: % of missing values in 6 pollutants

As proposed in the previous chapter, I will use K-Nearest Neighbor Imputation to fill in missing data and Min-Max scaling. For a missing AQI data points, it will be imputed by the mean of that AQI value of 3-most similar vectors (k=3), which have similar other AQI values and weather values. After using *KNNImputer* class from scikit-learn package, the result time series is shown as below:

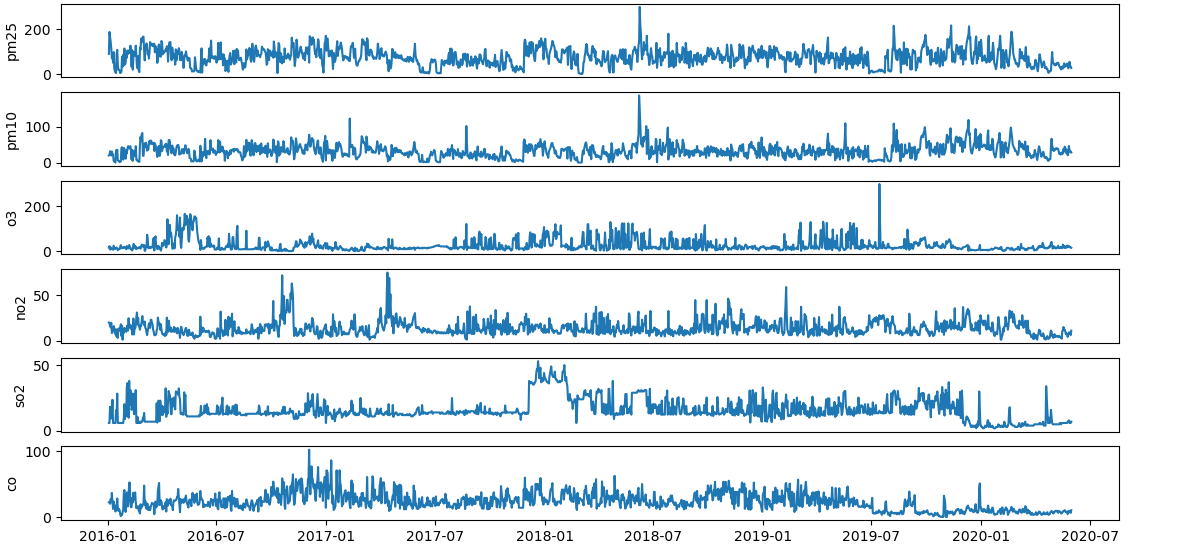


Figure .: Imputed AQI time series of 6 air pollutants

Summary information and the histogram of all 6 pollutants:

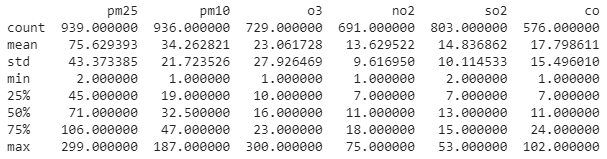


Figure .: Summary of each pollutants

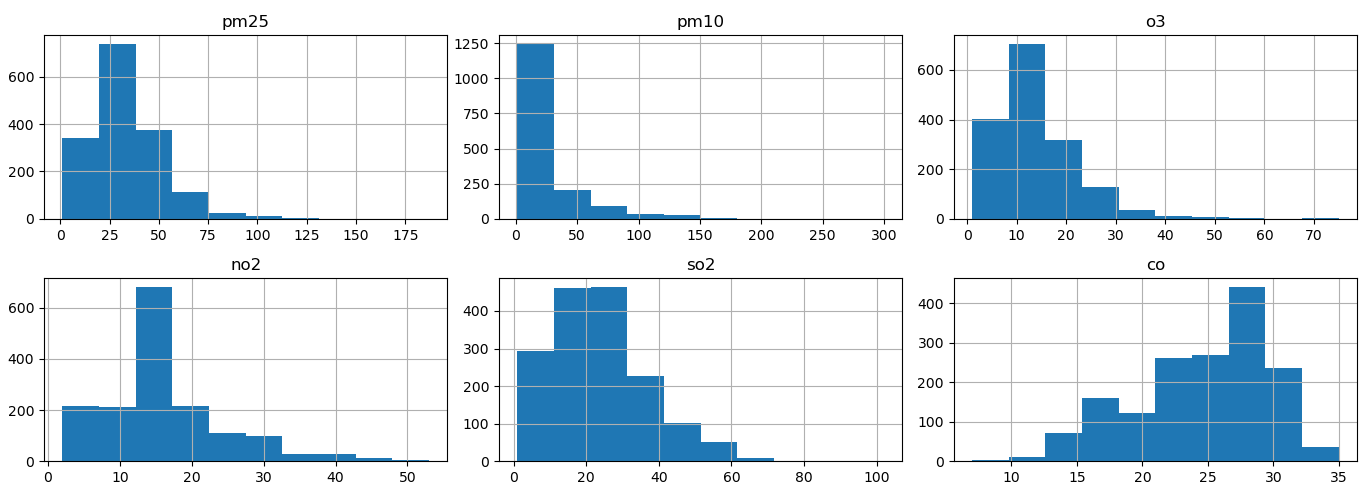


Figure .: A histogram for each pollutants

The count, mean, min, and max rows are self-explanatory. The std row shows the standard deviation, which measures how dispersed the values are. The 25%, 50%, and 75% rows show the corresponding percentiles: a percentile indicates the value below which a given percentage of observations in a group of observations fall. These are often called the 25th percentile (or first quartile), the median, and the 75th percentile (or third quartile). About histograms, many of them are tail-heavy: they extend much further to the right of the median than to the left. From that we can see there are many outliers found on these pollutants. But actually we can’t call them as outlier yet, instead they can be called as high unexpected value. Just imagine a day in a big city like Hanoi where dense air pollution might have occured due to heavy traffic from huge number of vehicles.

## Testing stationarity

Next, I use *statsmodels.tsa.stattools.adfuller* module to run Augmented Dickey-Fuller (ADF) test on all time series and check for its stationarity. The result as follows:

Table .: ADFuller Test result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PM2.5 | PM10 | O3 | NO2 | SO2 | CO |
| Test Statistic | -7.333 | -7.599 | -4.394 | -4.991 | -3.869 | -3.453 |
| Critical value 1% | -3.436 | -3.436 | -3.436 | -3.436 | -3.436 | -3.436 |
| Critical value 5% | -2.864 | -2.864 | -2.864 | -2.864 | -2.864 | -2.864 |
| Critical value 10% | -2.568 | -2.568 | -2.568 | -2.568 | -2.568 | -2.568 |
| P value | 0.0 | 0.0 | 0.0 | 0.0 | 0.002 | 0.009 |
| Result | Stationary | Stationary | Stationary | Stationary | Stationary | Stationary |

If a given p value is < significance level (0.05), we can reject the null hypothesis, which means the time series is stationary. As we can see, the p value of all tests is smaller than the significance level. The ADF test confirms all of the time series is stationary, so we do not have to difference them.

## Testing Granger’s Causality

In this step, I perform Granger’s Causality test on each pair of weather-AQI time series by using the module *grangercausalitytests* from package *statsmodels.tsa.stattools* and get the result as below:

Table .: Granger’s Causality Test result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Temperature | Pressure | Humidity | Wind speed |
| PM2.5 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| PM10 | 0.0001 | 0.0000 | 0.0000 | 0.0162 |
| O3 | 0.0348 | 0.0043 | 0.0167 | 0.3767 |
| NO2 | 0.0261 | 0.0004 | 0.0000 | 0.0311 |
| SO2 | 0.0355 | 0.0085 | 0.0001 | 0.0114 |
| CO | 0.0005 | 0.0000 | 0.0000 | 0.0410 |

If a given p value is < significance level (0.05), then the corresponding X series (column) causes the Y (row). For example, p value of 0.0001 at row 2, column 1 represents the p value of the Grangers Causality test for Temperature\_ causing PM­10\_ , which is less that the significance level of 0.05. So, we can reject the null hypothesis and conclude Temperature causes PM10. Looking at the above table, you can pretty much observe that almost the weather time series in the dataset are causing the pollutants, except for Wind speed is not causing O3 (0.3767), so I will leave out Ff (wind speed) featureswhen predicting O3.

## Vector Autoregression (VAR)

Now we are good to fit the data to the first model, which is Vector Autoregression (VAR) model. I will first divide the data set into 6 parts for each pollutants, each have the AQI values of each pollutants and 4 weather features (temperature, pressure, humidity, wind speed), for example:

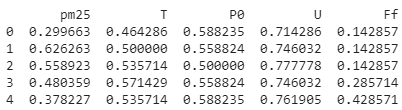


Figure .: Sub-data of PM2.5

The input data of VAR model will be the lagged values of the predicted AQI along with 4 weather features above. To select the right order of the VAR model, we iteratively fit increasing orders of VAR model and pick the order that gives a model with least AIC. The result is below:

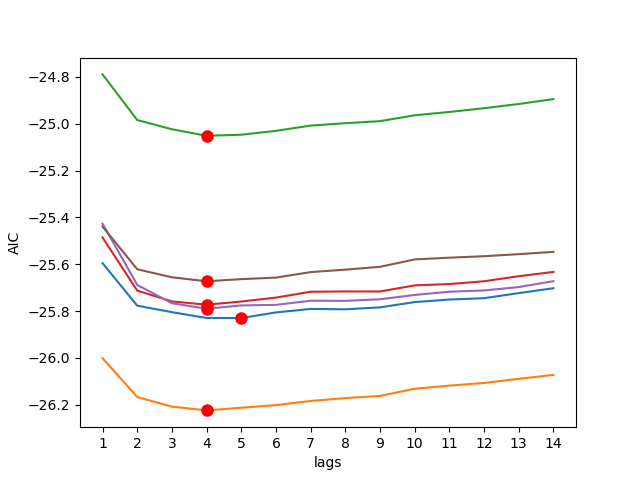


Figure .: AIC of differents lag order

In the above output, in 5 out of 6 model for 6 time series, the AIC drops to lowest at lag 4, so the lag 4 model should be selected. The input data of the model is the values of the AQI and the weather in the previous 4 days. For example, to predict the AQI of PM2.5, the following fourth order VAR(4) model is used:

|  |  |  |
| --- | --- | --- |
|  |  | Equation . |

Next I start training all 6 time series and obtains MAE(Mean Absolute Error) loss as a measure of the accuracy of VAR model:

Figure .: MAE loss of VAR model

Finally, by using the test set, the actual AQI values are compared with the predicted values of the next 1 day generated by VAR model. The blue line represents the actual time series, the red line represents the predicted time series. MAE is also the metric to evaluate the forecasts:

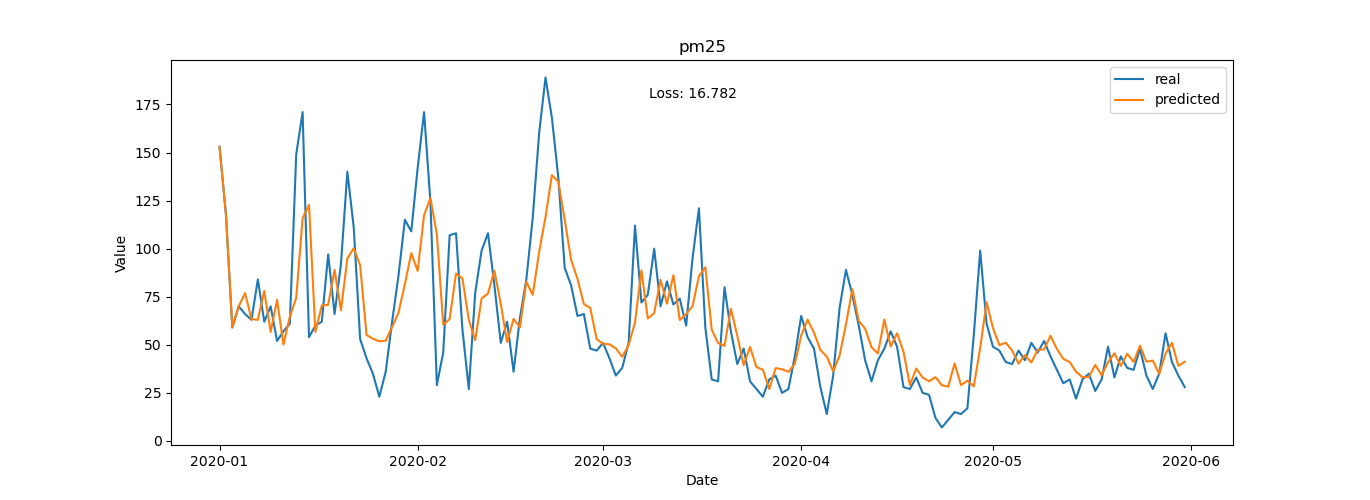


Figure .: Actual vs predicted PM2.5 of the next 1 day (VAR)

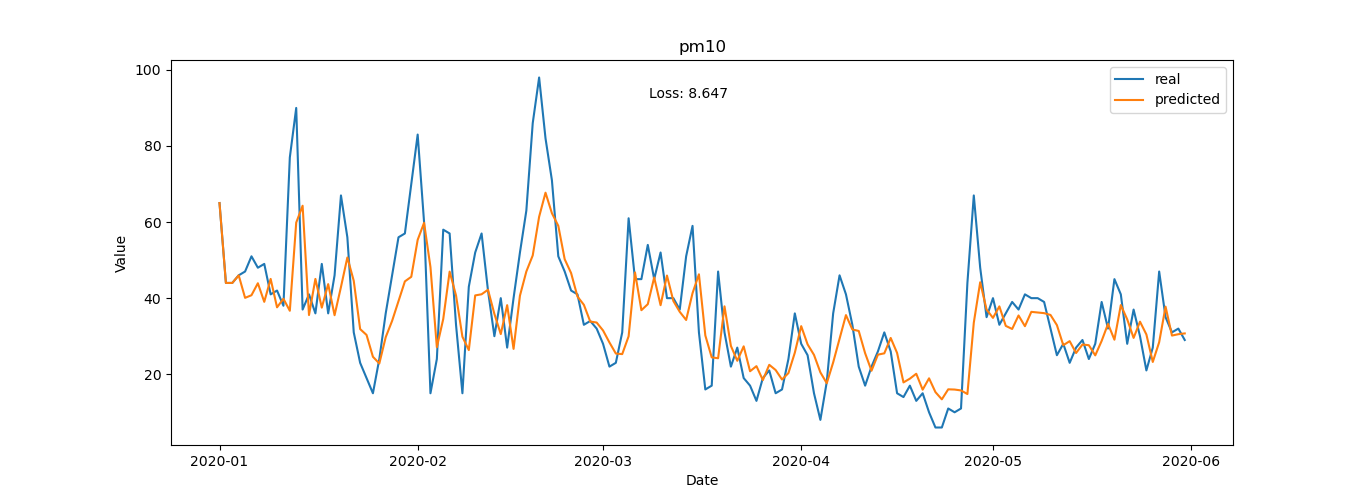


Figure .: Actual vs predicted PM10 of the next 1 day (VAR)

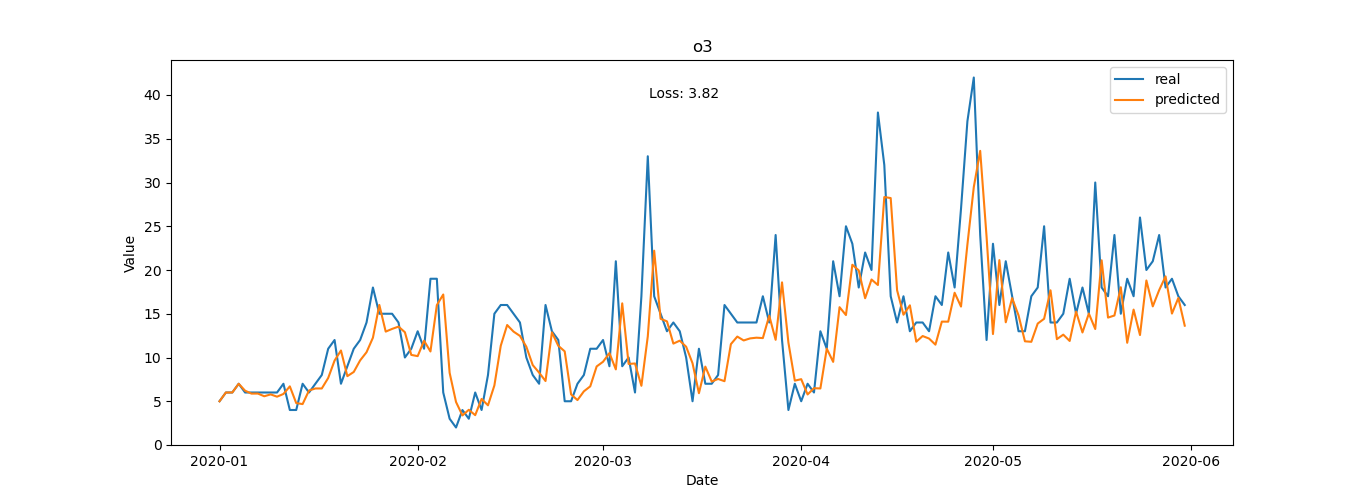


Figure .: Actual vs predicted O3 of the next 1 day (VAR)

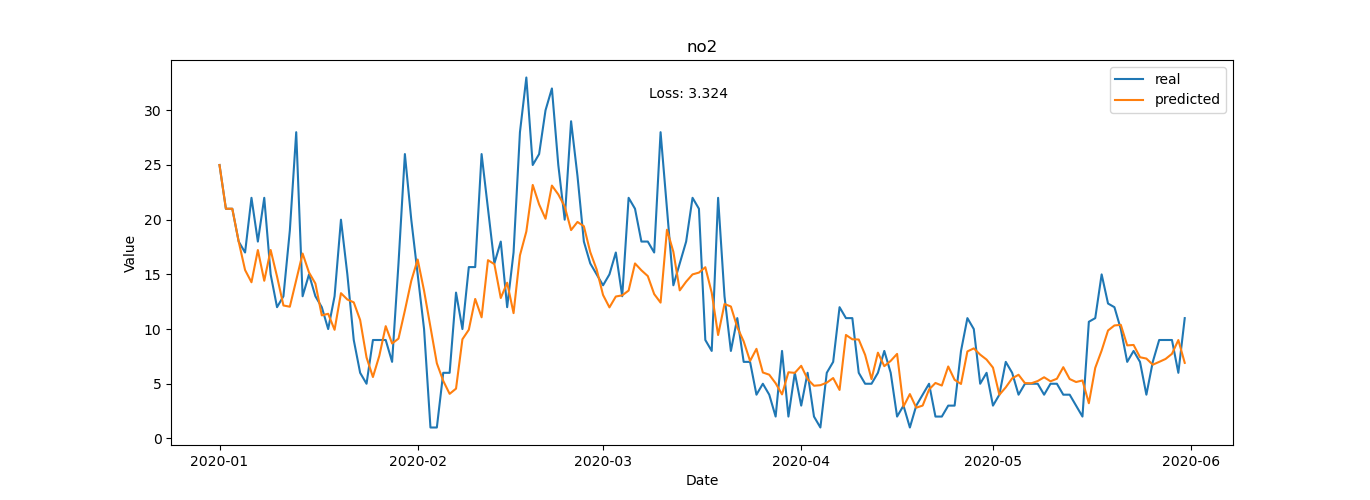


Figure .: Actual vs predicted NO2 of the next 1 day (VAR)

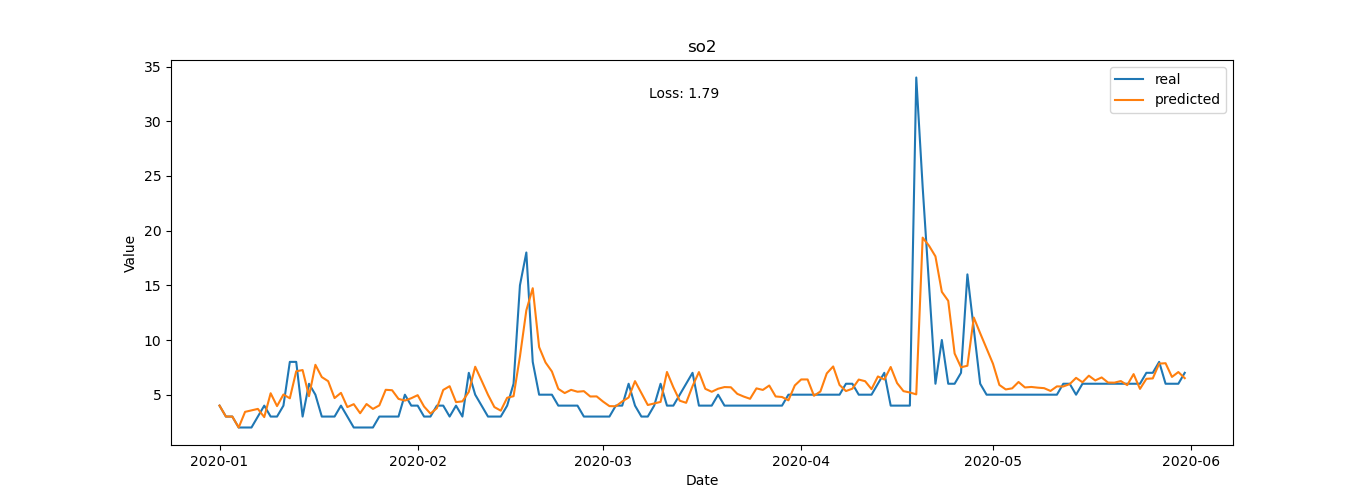


Figure .: Actual vs predicted SO2 of the next 1 day (VAR)

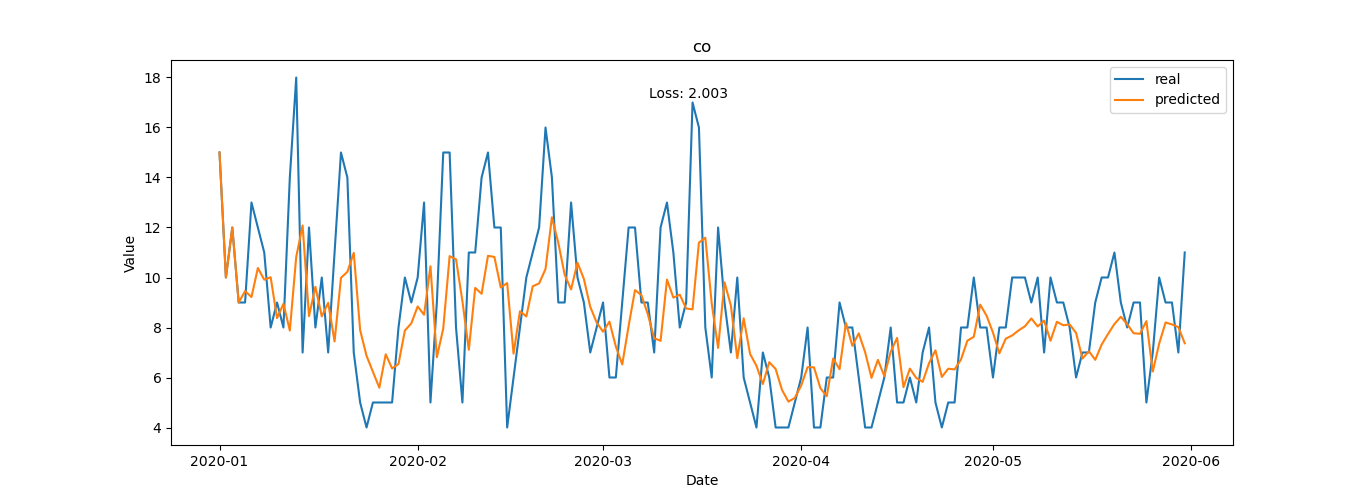


Figure .: Actual vs predicted CO of the next 1 day (VAR)

I also compare the actual and predicted AQI values of the next 3 day:

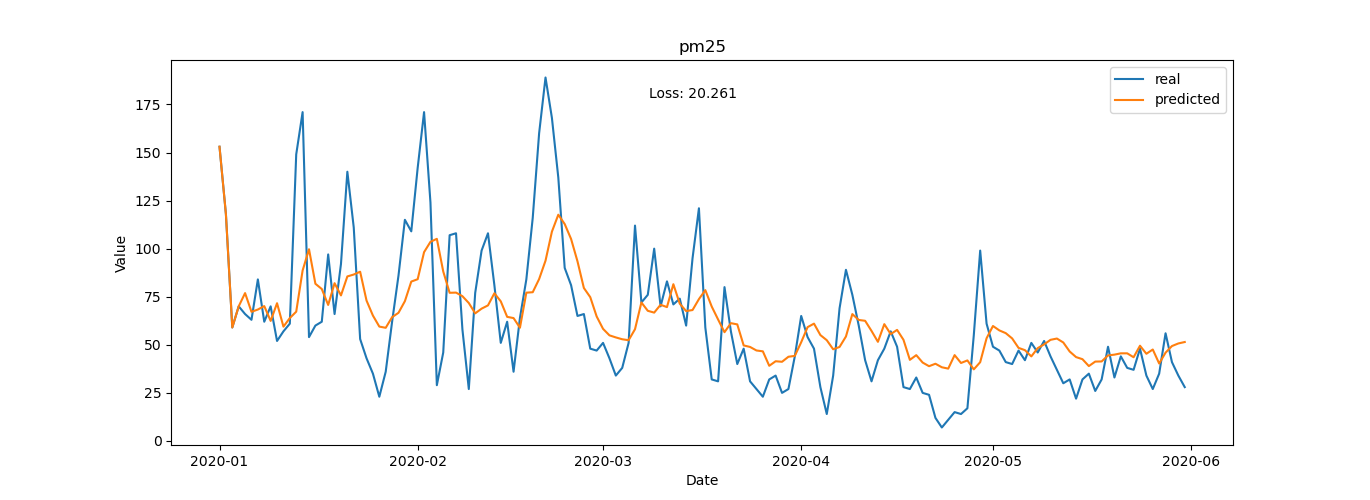


Figure .: Actual vs predicted PM2.5 of the next 3 days (VAR)

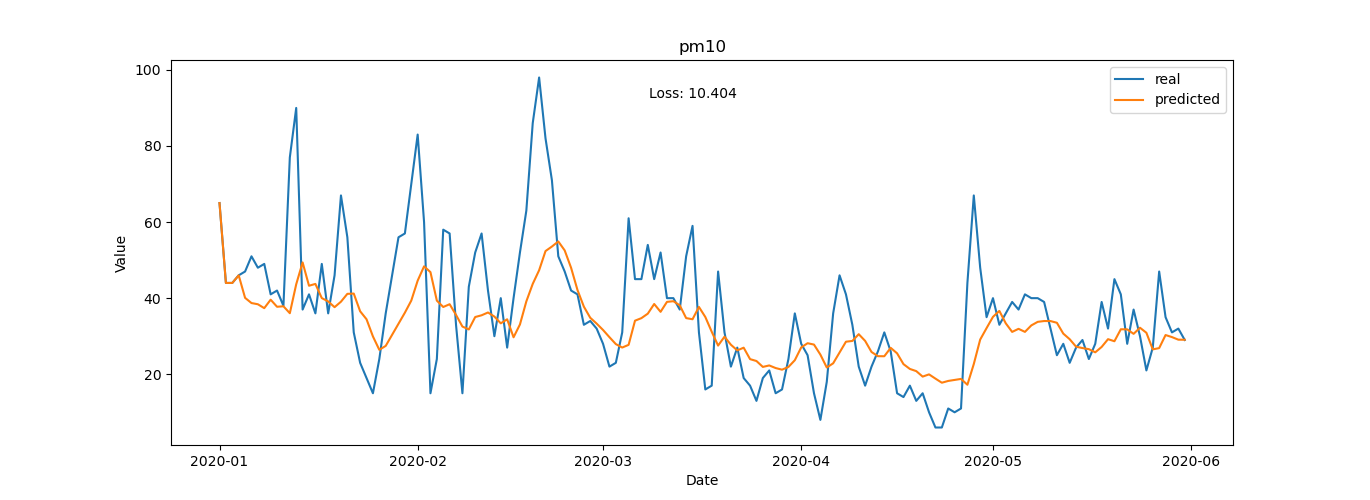


Figure .: Actual vs predicted PM10 of the next 3 days (VAR)

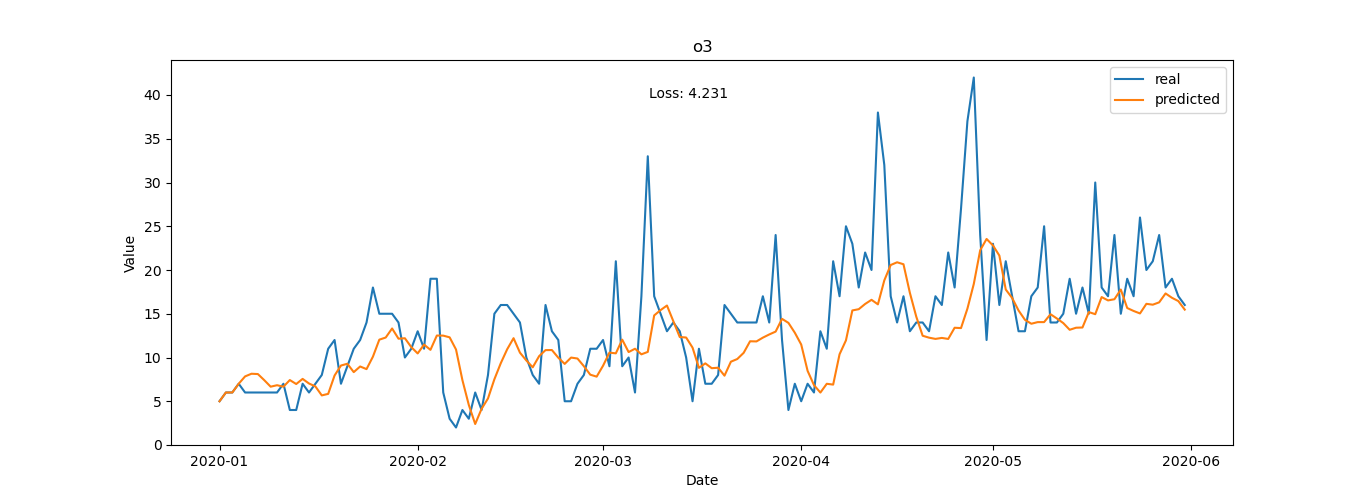


Figure .: Actual vs predicted O3 of the next 3 days (VAR)

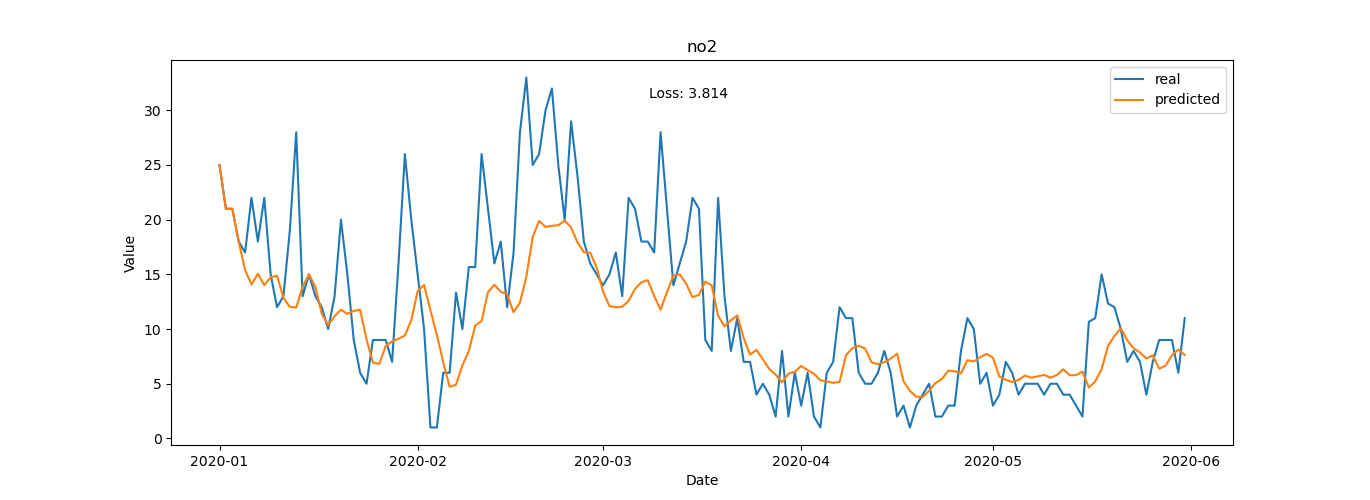


Figure .: Actual vs predicted NO2 of the next 3 days (VAR)

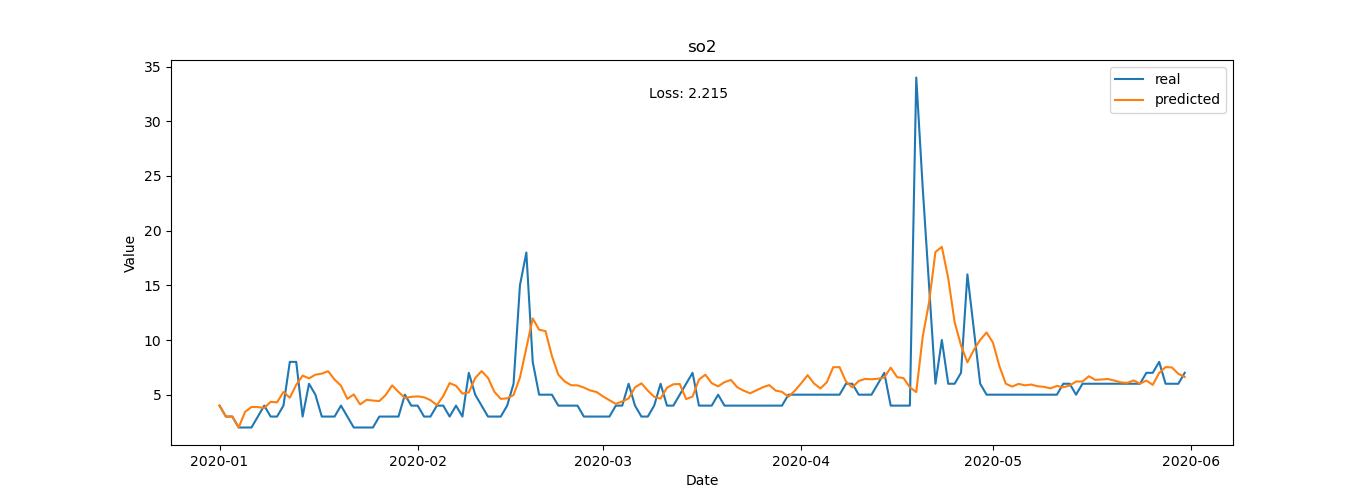


Figure .: Actual vs predicted SO2 of the next 3 days (VAR)

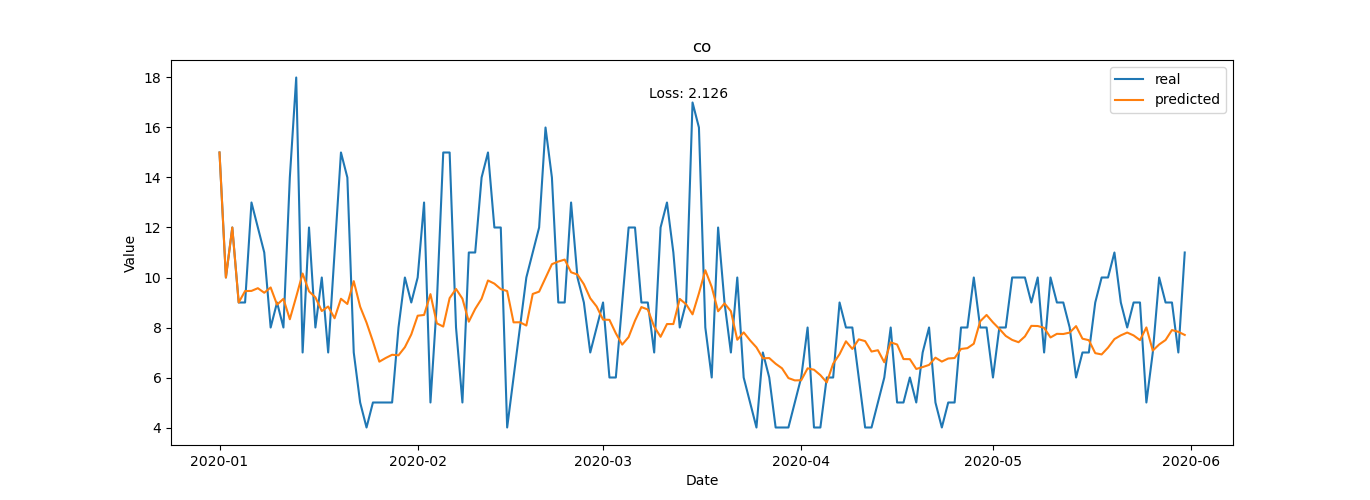


Figure .: Actual vs predicted CO of the next 3 days (VAR)

## Long Short-Term Memory

Before we can train a neural network, we need to model the data in a way the network can learn from a sequence of past values. Specifically, LSTM expects the input data in a specific 3D tensor format: [samples, timesteps, features]. As a supervised learning approach, LSTM requires both features and labels in order to learn. In the context of time series forecasting, it is important to provide the past values as features and future values as labels, so LSTM’s can learn how to predict the future. Therefore, we explode the time series data into a 2D array of features, where the input data consists of overlapping lagged values of meteorological data at the desired number of timesteps in batches. I generate a 1D array consisting of only the labels or future values of AQI which we are trying to predict for every batch of input feature. The input data also should include lagged values of the AQI so the network can also learn from past values of the labels.

To compare the results with VAR model, I also choose the data of 4 days in the past to predict values some time in the future. The input data is a 2D arrays of 20 features (5x4), and in case I want to predict 1 day-ahead values, the data will be as follows:

Table .: Supervised form of data to predict 1 day-ahead AQI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PM2.5(t-4) | T(t-4) | P0(t-4) | U(t-4) | Ff(t-4) | PM2.5(t) |
| PM2.5(t-3) | T(t-3) | P0(t-3) | U(t-3) | Ff(t-3) |
| PM2.5(t-2) | T(t-2) | P0(t-2) | U(t-2) | Ff(t-2) |
| PM2.5(t-1) | T(t-1) | P0(t-1) | U(t-1) | Ff(t-1) |

The data will look something like this:

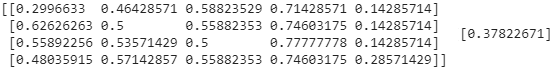


Figure .: Supervised data for LSTM

For the case predicting 3 day-ahead values, the input data remain the same but the labels will contain 3 AQI values of the next 3 days:

Table .: Supervised form of data to predict 3 day-ahead AQI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PM2.5(t-4) | T(t-4) | P0(t-4) | U(t-4) | Ff(t-4) | PM2.5(t) |
| PM2.5(t-3) | T(t-3) | P0(t-3) | U(t-3) | Ff(t-3) | PM2.5(t+1) |
| PM2.5(t-2) | T(t-2) | P0(t-2) | U(t-2) | Ff(t-2) | PM2.5(t+2) |
| PM2.5(t-1) | T(t-1) | P0(t-1) | U(t-1) | Ff(t-1) |  |

Like mentioned above, the input data should be reshaped to 3D format: [n\_samples, 4, 5] before fitted to the model.

Now it’s time to create the LSTM model. I will use stacked LSTM architecture, which comprised multiple hidden LSTM layers stacked on top of each other. For many problems, you can start with just one or two hidden layers and the LSTM network can model even the most complex functions, provided it has enough neurons. I will choose 2 hidden layers, an LSTM layer with 20 neurons above provides a sequence output rather than a single value output to the LSTM layer with 20 neurons below:

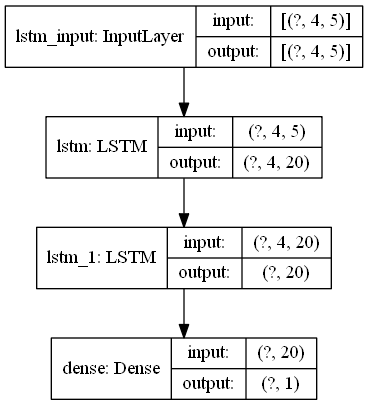


Figure .: LSTM model architecture

After experimenting multiple times, I have found the hyperparameters which have the best results:

* Epochs (Number of times all the data will be passed to the neural network): 100 epochs, with EarlyStopping callback to interrupt training when it measures no progress on the validation set for a number of epochs
* Batch size (number of samples data will be passed to the neural network before updating the model parameters): 16 samlples in one batch
* Optimizer: Adam (adaptive moment estimation) converge faster than GD or SGD, and might be able to converge to better solutions
* Loss function: Mean Absolute Error

The training loss and validation loss after fitting the data to LSTM model to forecast AQI values of the next 1 day is shown as below:

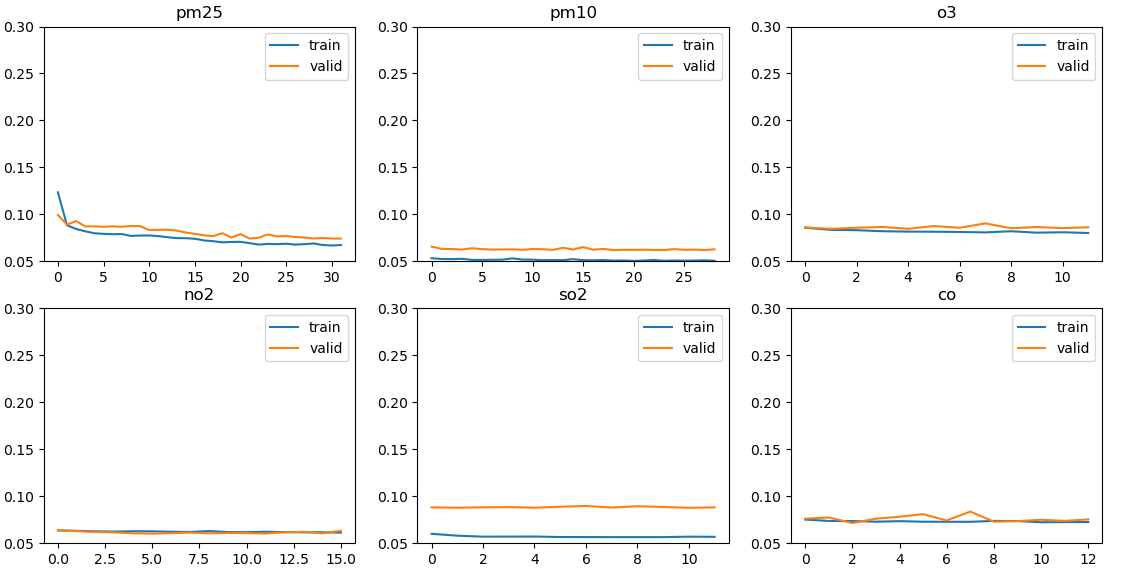


Figure .: Loss during traning of 6 time series

We can see that in most cases, the training loss is appoximately equal to the validation loss, which is a good sign. Except for SO2, the training loss is lower than validation loss, which means the network might be overfitting a little bit. Then I will compare the actual values with the predicted values:

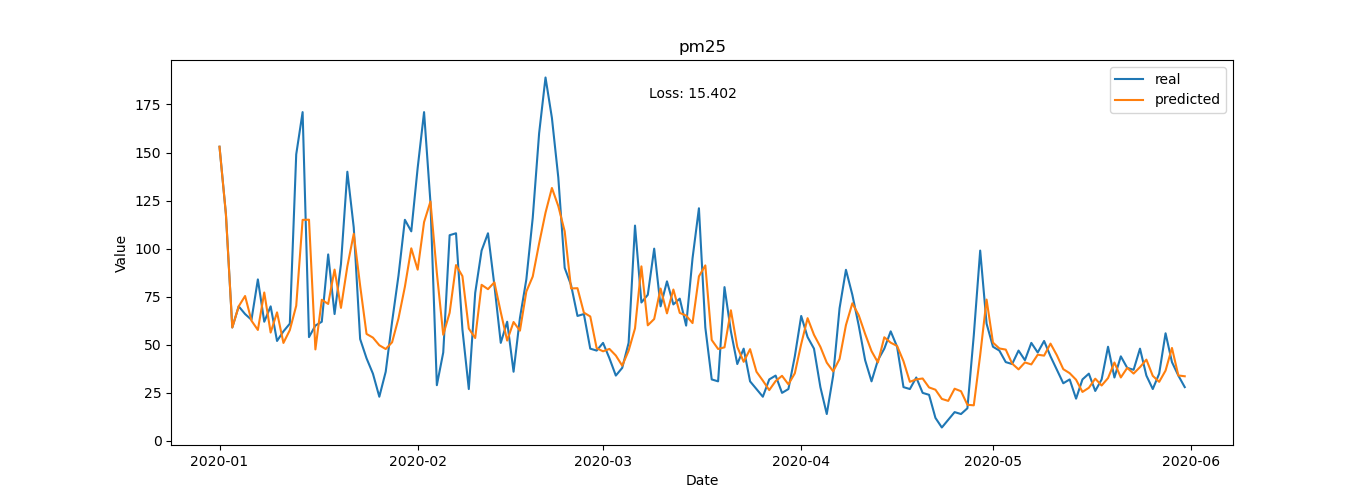


Figure .: Actual vs predicted PM2.5 of the next 1 day (LSTM)

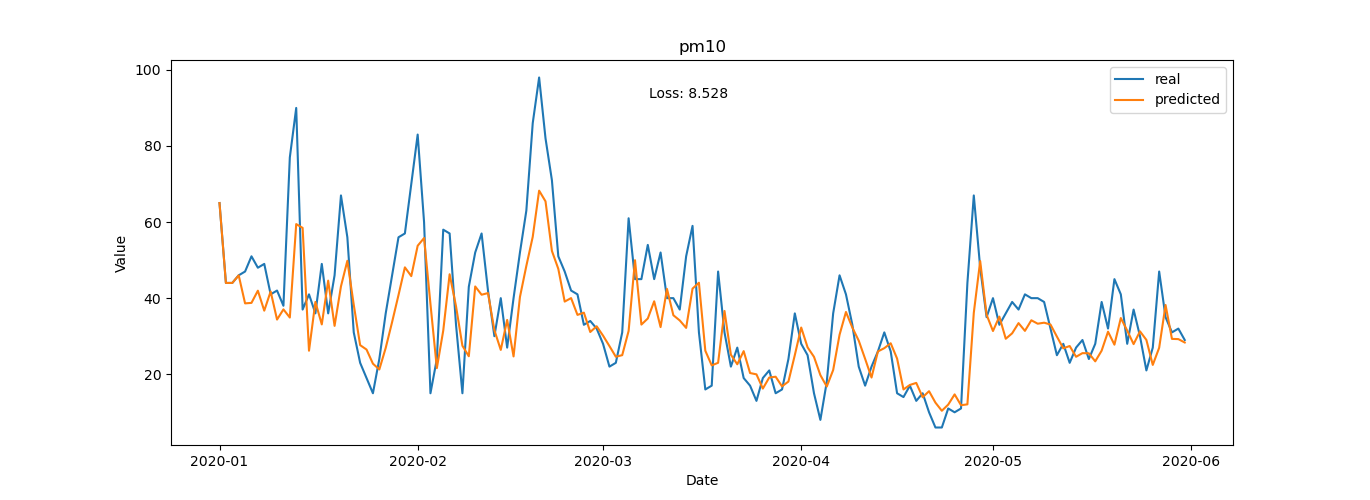


Figure .: Actual vs predicted PM10 of the next 1 day (LSTM)

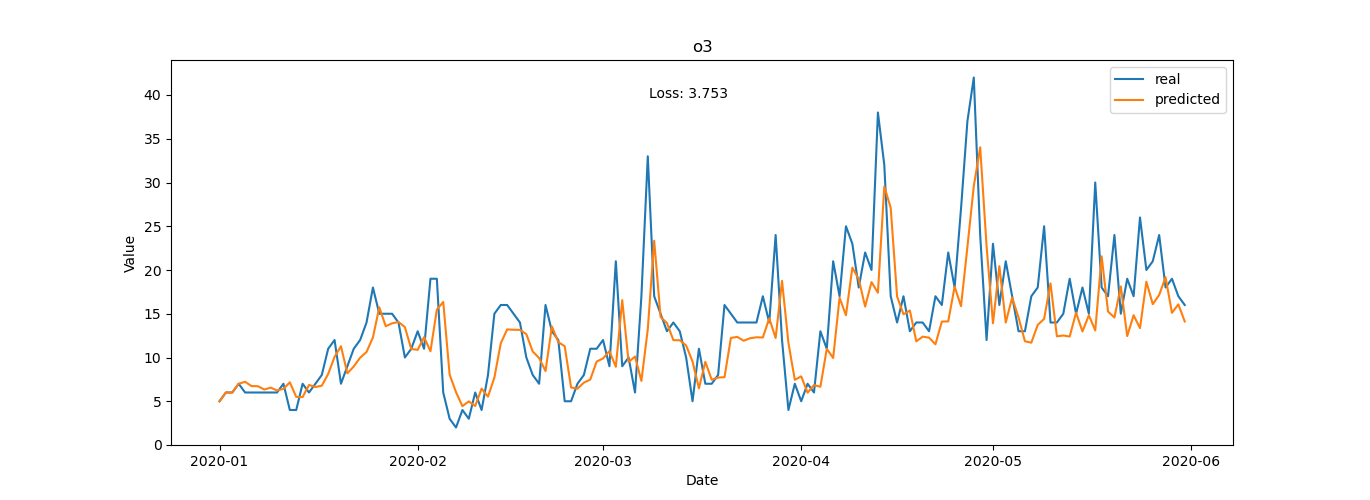


Figure .: Actual vs predicted O3 of the next 1 day (LSTM)

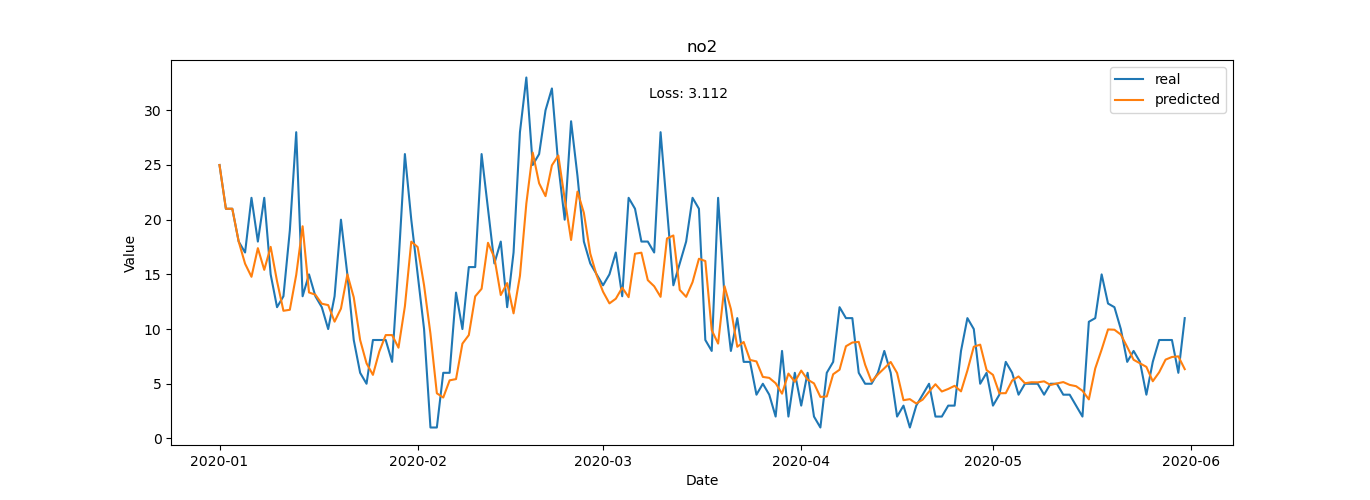


Figure .: Actual vs predicted NO2 of the next 1 day (LSTM)

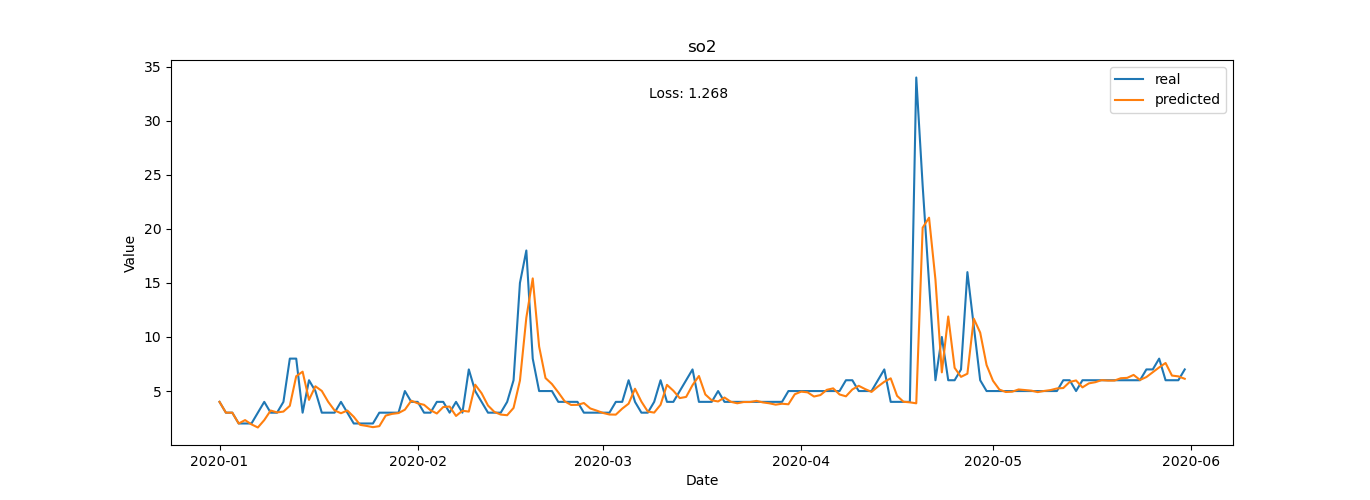


Figure .: Actual vs predicted SO2 of the next 1 day (LSTM)

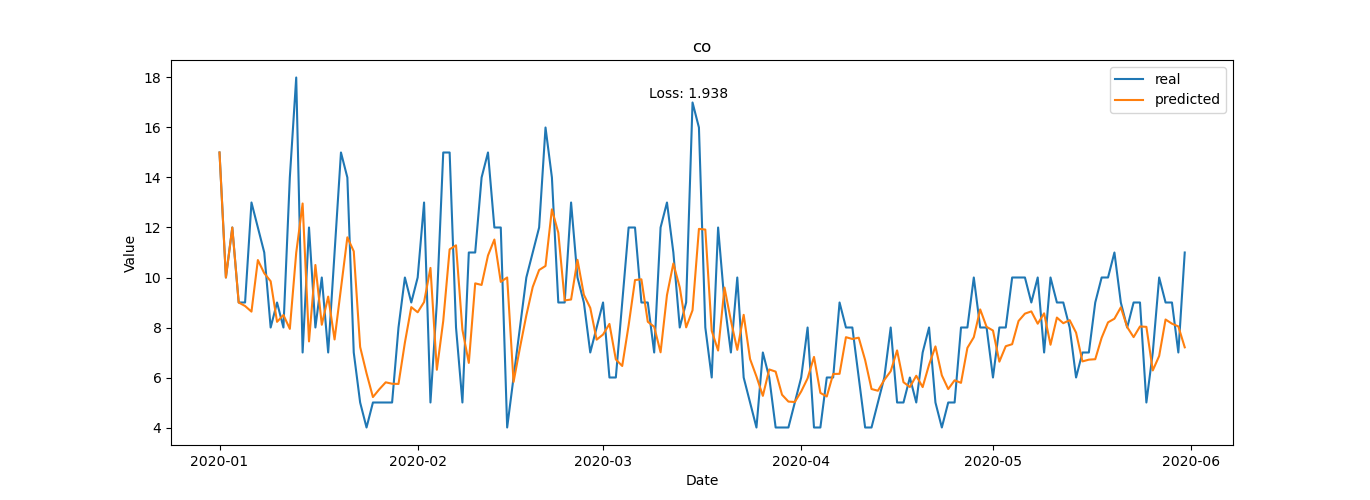


Figure .: Actual vs predicted CO of the next 1 day (LSTM)

Continue with the case of predicting AQI value for the next 3 days, I train the LSTM network again and obtain the training loss and validation loss as follows:

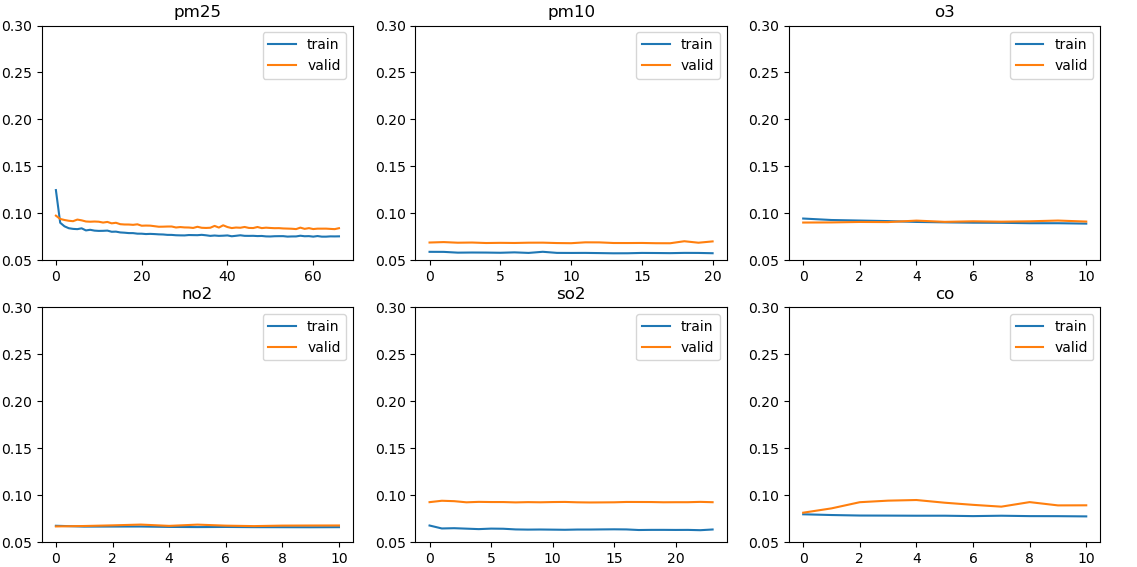


Figure .: Loss during traning of 6 time series

Both losses are slightly increased compared to the 1-day forecasting case but the pattern remained the same. Finally, the actual values is, again, compared with the predicted values:

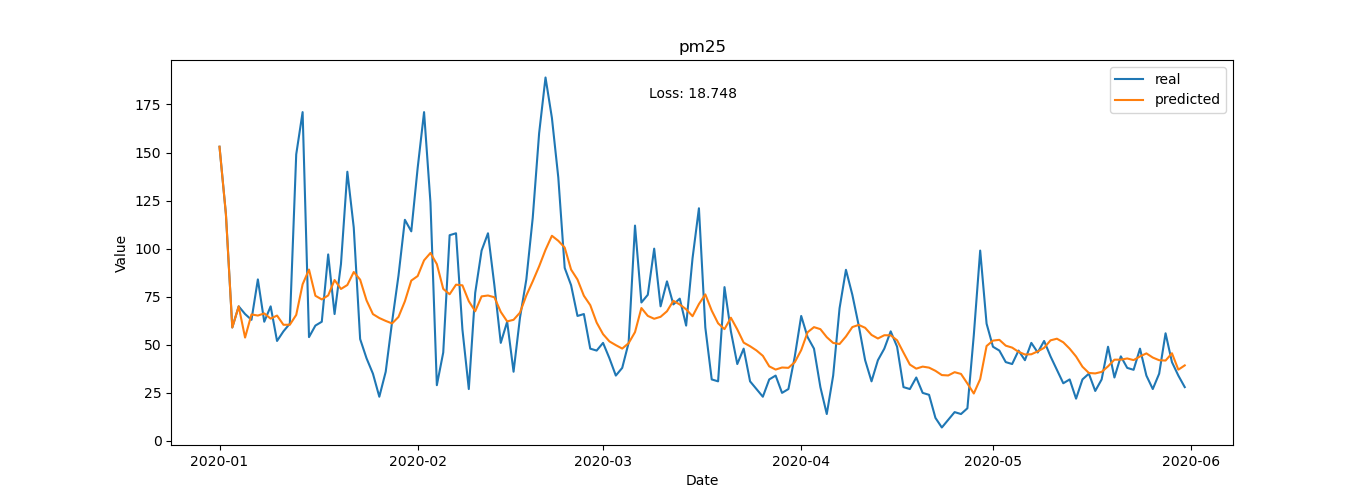


Figure .: Actual vs predicted PM2.5 of the next 3 days (LSTM)

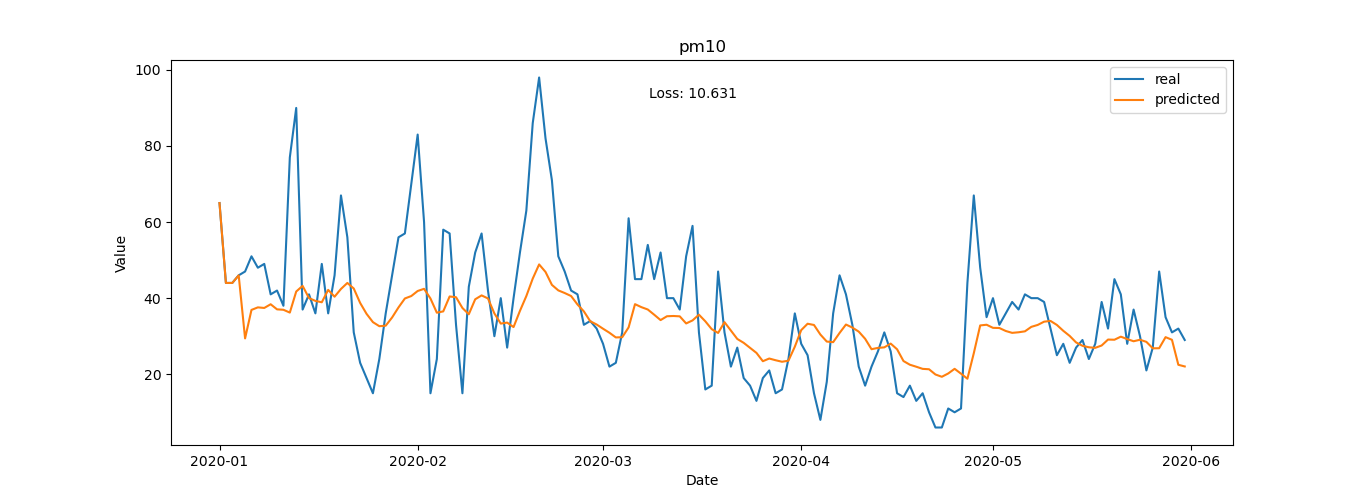


Figure .: Actual vs predicted PM10 of the next 3 days (LSTM)

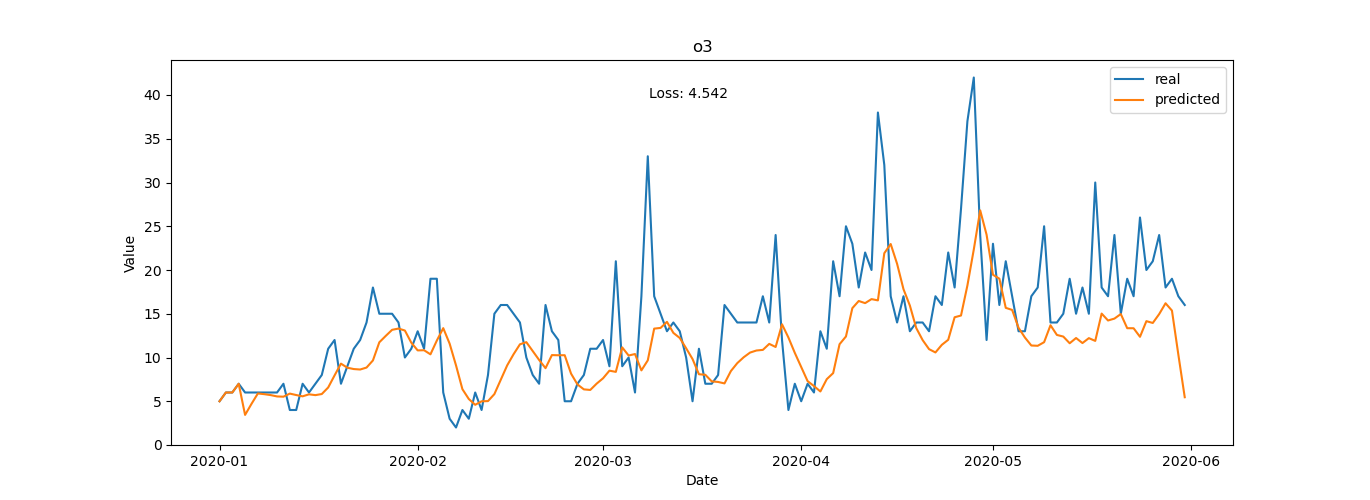


Figure .: Actual vs predicted O3 of the next 3 days (LSTM)

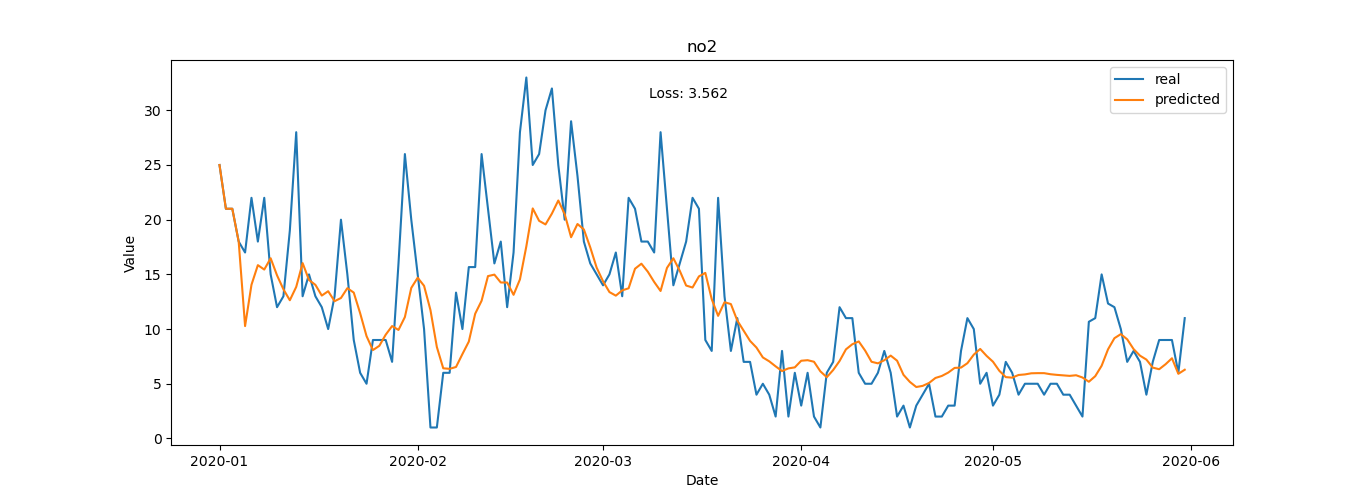


Figure .: Actual vs predicted NO2 of the next 3 days (LSTM)

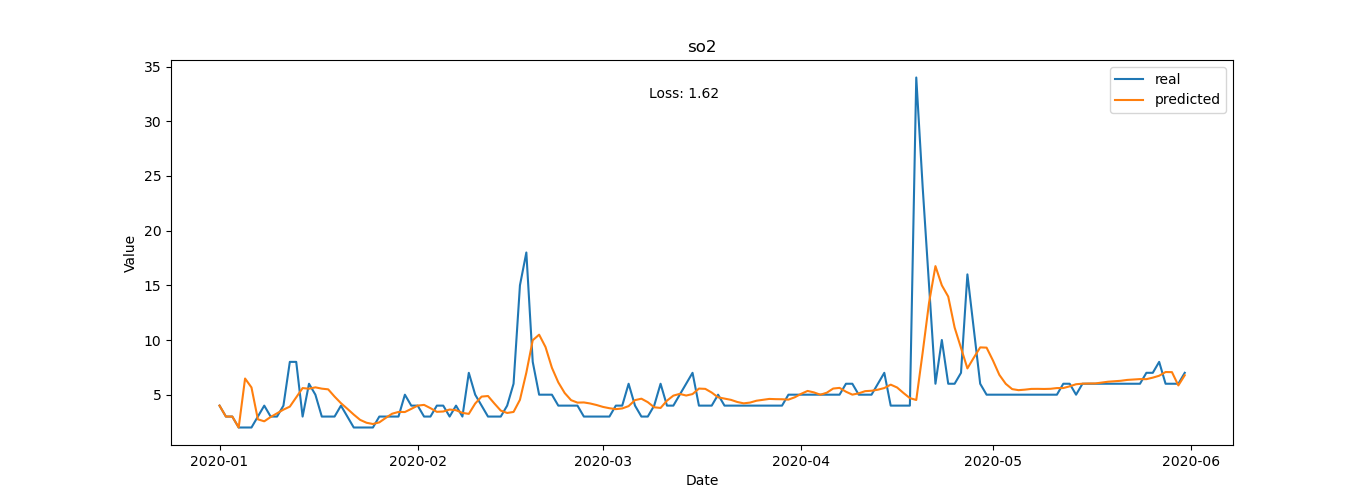


Figure .: Actual vs predicted SO2 of the next 3 days (LSTM)

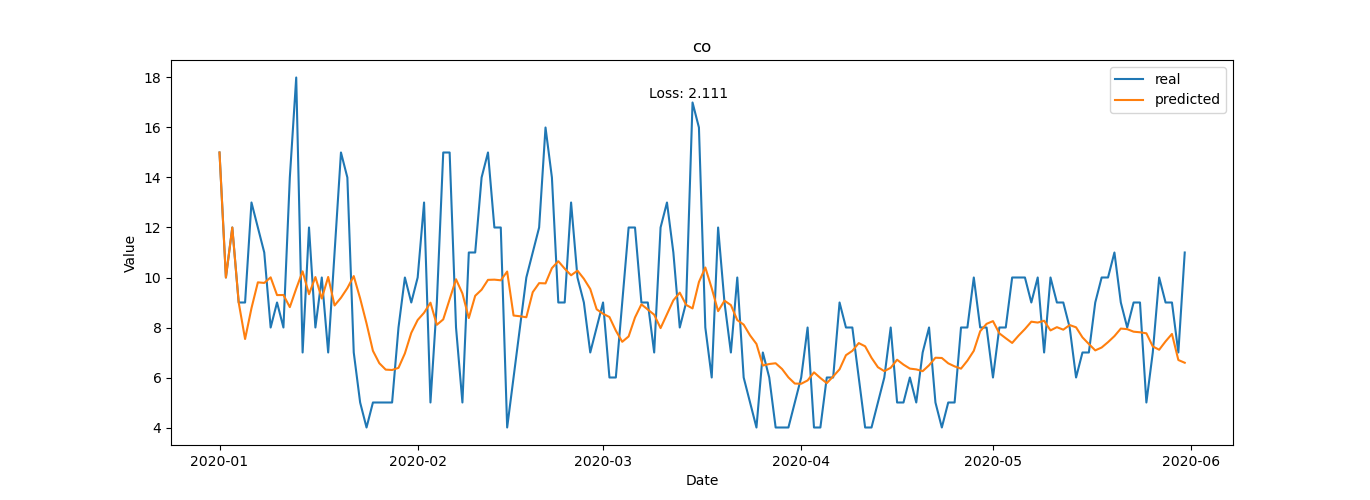


Figure .: Actual vs predicted CO of the next 3 days (LSTM)

## Result summary

Look at all the figures above, I summarized the loss in the evaluate phase in the following table:

Table .: MAE loss of the models when comparing actual and predicted values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | VAR(1-day) | LSTM(1-day) | VAR(3-day) | LSTM(3-day) |
| PM2.5 | 16.782 | 15.402 | 20.261 | 18.748 |
| PM10 | 8.647 | 8.528 | 10.404 | 10.631 |
| O3 | 3.779 | 3.753 | 4.231 | 4.542 |
| NO2 | 3.324 | 3.112 | 3.814 | 3.562 |
| SO2 | 1.79 | 1.268 | 2.215 | 1.62 |
| CO | 2.003 | 1.938 | 2.126 | 2.111 |

In general, for the 1-day forecast problem, both models can capture the pattern of the time series quite well and correctly forecast the trend of the series. LSTM model gives slightly better results than VAR model in predicting AQI values for the next 1 day. The prediction for SO2 is closest to the real value. The predictions for PM10 and PM2.5 are actually not too bad, but the loss is high because the AQI values of these two pollutants are large compared to others. For the prediction over the next 3 days, the results of both models are a little worse and neither yields better results in all series.

In summary, the predicted values are only approximate and serve for this particular study because the air quality prediction problem is complicated and requires much more data.

# CONCLUSION

## Conclusion

## The thesis with the topic “Analysis and prediction of air quality data” is basically completed and performed the following tasks:

## Introduce to the forecasting task, get an overview of the air quality problem in big cities like Hanoi, learn about some of the main air pollutants to better understand its effect on human health, have some basic knowledge about the Air Quality Index (AQI) and show the impact of meteorology to air quality.

## Determining the input data for the problem and analyzing, preprocessing the input data. From there, two theoretical methodologies have been proposed using Vector Autoregression model and Long Short-Term Memory network to predict the future AQI values of 6 pollutants: PM2.5, PM10,O3, NO2, SO2, CO.

## Implement the prediction pipeline in Python language for the two methods proposed above. Evaluate the results of these models and compare them with each other.

Because this is just an experiment about the air quality prediction problem, so I encountered many difficulty. The main difficulty in making this thesis is the difficulty in finding data for the experimental implementation. The longer the historical data, the better for the models to extract the information needed to make effective predictions. Because the method requires historical data for such a long period, manual collection is difficult. The dataset collected in this thesis is still lacking and have many missing data points, so the prediction is not completely accurate.

The next difficulty is in developing the LSTM network. The model has quite a lot of different architectures and hyperparameters, so training all these different variants is time consuming. The model proposed in this thesis is still quite simple and may not be the best one.

## Direction for further research

There are two important research directions for the future: collect furthermore datasets and choosing better algorithms. As mentioned above, data plays an important role in any forecasting method so it is necessary to collect as much data as possible for model training. Besides, more advanced models developed by researchers and scientists can be experimented and compared with the current results to yield better prediction.

# REFERENCES

|  |  |
| --- | --- |
| [1] | WHO, "WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide," 2005. |
| [2] | J.P.Shi, "Regression modelling of hourly NOx and NO2 concentrations in urban air in London," 1997. |
| [3] | M. S. K. Abhilash, "Time Series Analysis of Air Pollution in Bengaluru Using ARIMA Model," 2018. |
| [4] | B.-C. Liu, "Urban air quality forecasting based on multi-dimensional collaborative Support Vector Regression (SVR): A case study of Beijing-Tianjin-Shijiazhuang," 2017. |
| [5] | V. R. Prybutok, "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations," 2000. |
| [6] | A. Vlachogianni, "Evaluation of a multiple regression model for the forecasting of the concentrations of NOx and PM10 in Athens and Helsinki," 2011. |
| [7] | Y.-T. Tsai, "Air pollution forecasting using RNN with LSTM," 2018. |
| [8] | R. Navares, "Predicting Air Quality with Deep Learning LSTM: Towards Comprehensive Models," 2019. |
| [9] | R. J. Hyndman, Forecasting: Principles and Practice, 2013. |
| [10] | "Air Quality Historical Data Platform," [Online]. Available: https://aqicn.org/data-platform/register/. |
| [11] | "Reliable Prognosis," [Online]. Available: rp5.ru. |