AIR POLLUTION
Beijing

1. Introduction

During the past six decades, a growth in pollution has been caused by a number of factors, including urbanisation, industry, cars, power plants, chemical operations, crop fires, volcanic eruptions, and wildfires. All of these factors have contributed to an increase in pollution. Particulate matter, often known as PM, is one of the primary factors that leads to air pollution, and each of these actions contributes to the accumulation of PM (Jung, 2005) . There are a number of harmful gases and other substances floating around in the air, all of which are contributors to the problem of air pollution. One of the dangerous compounds that can be found in the air is called PM, which stands for particulate matter. A high value for this statistic may be associated with increased levels of air pollution. Recent years have seen a rise in awareness of air pollution's status as a major global issue and a significant contributor to mortality rates. According to the World Health Organization (WHO), air pollution is responsible for 6.9 million fatalities each and every year around the globe. Additionally, it has been determined that PM is the fourth biggest cause of death around the world (Soh, 2015). Every year, thousands of individuals all around the world lose their lives due to air pollution. Taiwan and India, two of the most populous nations in the world, have the greatest fatality rates associated with chronic illnesses and asthma, respectively. Air pollution puts the lives of a great number of people at danger, particularly the lives of the fifty percent of youngsters who are harmed by it (Mehdipour, 2015). Particles and various forms of contaminants, such as PM2.5, SO2, NOx, CO, PM10, and O3, have a close association. The amount of pollution that is present in the air as well as the particle size distribution are both described here. PM2.5 particles, which have a mass that is less than 2.5 micrograms, have been related to a wide variety of health problems, including those relating to the cardiovascular system and the respiratory system. As a direct consequence of this, there is a significant impact on one's state of health. At this point in time, the establishment of a structure for the reporting of the air quality by average citizens is regarded as a crucial requirement (Delavar, 2008)**.** It is possible for sensors to fail, which might lead to difficulties in data collection. In this area, an algorithm for evaluating quality of the air in smart urban has a lot of promise.

We found that a few of the locations did not have any kind of continuous monitoring mechanism for the air quality. The same can be said for cities that are located in close proximity to significant businesses. It is concerning to note that if the authorities are unaware of the quantity of particle matter PM that is present in the air, then they will undoubtedly be ignorant of the quality of the air. This is a really concerning possibility (Soh, n.d.). Is it better or worse for our health to breathe this air? As a direct consequence of this, they have decided not to take any action in response to it. As a consequence of this, I devised a method for forecasting the quality of the air based on a variety of parameters.

The primary objective of this investigation is to evaluate the performance of various different deep learning strategies for particle prediction (PM2.5). As a result of this, this model is able to forecast the occurrence of sensory failure in real time, even when accounting for extremely minute differences in particle size (PM2.5). In this paper, these deep learning models are primarily focused on PM2.5 as their target variable. Kaggle also provides the data that is necessary to train the deep learning models that are being used. The primary objective of this project is to evaluate a deep learning model for the prediction of particulate matter (PM) in the atmosphere. At last, the models are judged against one another to determine which one is superior. The LSTM model is the one that we have finally decided to go with (C. Brokamp, 2017). The first step in developing these models and putting them to the test will be performing an analysis of the data.

2. Background

Industrialization has increased in recent years to better the lives of people. This has a severe effect on the ecology as well, causing global warming and pollution of the atmosphere. Water contamination created by industrial waste is not the only pollutant that impacts the environment. One or more of these contaminants can impair human health. In addition, it has been associated to cognitive abnormalities in youngsters (Johnston, 2021). Climate change and air pollution projections are crucial components of a smart city's plan for making informed decisions and taking effective action (Sun, 2021) . Thus, the application of deep learning in automated prediction systems is becoming more frequent. The development of more exact methods for detecting pollution levels in the air has been the focus of various studies in recent years. Deterministic and statistical methods are employed to estimate air pollution concentrations in general (X. Xi, 2015)**.**

As a result, air quality predicting has emerged as an important scientific topic. The usage of Artificial Neural Networks (ANN) is consistent with internal operations that take into account air pollution hypothesis, as contrasted to other models such as multiple linear regression. The ANNs display some of the same problems as this team, such as isolation , because this is a frequent approach (F. Zhang, 2014).

In order to offer the most accurate evidence of the show's judgement, a lengthy procedure is required. As far as air pollution detection and prediction go, it's worth highlighting those contemporary systems integrate mechanical learning approaches with the usage of air pollution data (Wang, 1999). In addition, LSTM is commonly used to predict air pollution. This project will also investigate LSTM.

3. Objective

The main purpose of this investigation is to evaluate the Long-Short-Term Memory (LSTM) model by making use of sequential data that is associated with the quality of the air. We are aware that the data on air quality is presented in a sequential format, and that both conventional machine learning models and deep learning models are unable to deal with this format. Even when processing data in a sequential fashion, the LSTM's combination of long-term and short-term memory units can lead to excellent performance. Because of this, the purpose of this work is to investigate whether or not LSTM is capable of processing sequential data.

4. Methodology

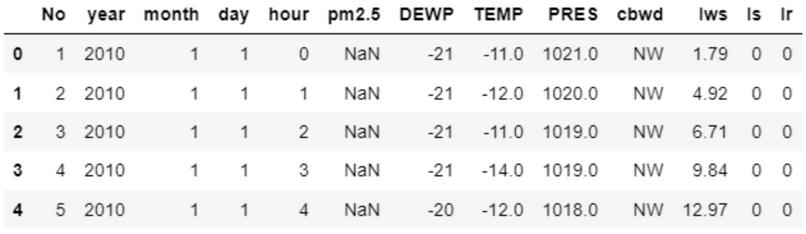
When working on a machine learning project, our team always follows the same set of procedures, and those procedures make up the methodology for this particular project. These strategies are broken down into even more specific steps below:

4.1. Dataset

While operating a machine learning or deep learning project, data becomes the important and basic component of the project. Because machine learning models are trained on historical data, or more precisely, data that has already been acquired, and then used to generate predictions about data that has yet to be collected. We can't teach them to predict the future until we have data (Ding, J. Zhang and W., 2017). Therefore, we required some sort of working dataset in order to put our LSTM and ANN models through their paces. There were a large number of datasets that were made available after being compiled by sensors located in a variety of locations. For example, some organizations collected data in Taiwan while others did it in India; these datasets are now freely available on a variety of data websites, including Kaggle, UCI, and others (Liu, 2012) **.** Therefore, I used the dataset that was related to the PM 2.5 index in Beijing, which is located in China.

This dataset was gathered in Beijing, the capital of China, by a variety of sensors located in a variety of locations; however, they did not put the whole and original data as it was collected on the data sites (X. Ni, 2017). They made minor modifications to the dataset in order to conceal sensitive information, but they kept the primary data that was connected to the PM 2.5 index. On Kaggle, you may find this particular dataset. It is visible in this location.

Here is first look of data:

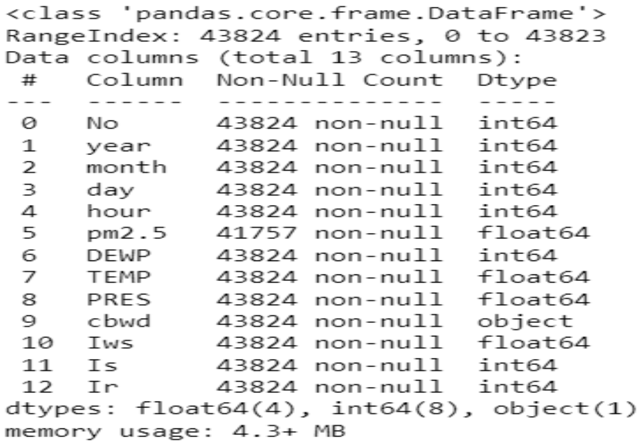


	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
0	1	2010	1	1	0	NaN	-21	-11.0	1021.0	NW	1.79	0	0
1	2	2010	1	1	1	NaN	-21	-12.0	1020.0	NW	4.92	0	0
2	3	2010	1	1	2	NaN	-21	-11.0	1019.0	NW	6.71	0	0
3	4	2010	1	1	3	NaN	-21	-14.0	1019.0	NW	9.84	0	0
4	5	2010	1	1	4	NaN	-20	-12.0	1018.0	NW	12.97	0	0

Figure 1: First look of data

4.2.Basic Description about data

The information contained in this dataset is organised into 13 columns and 43824 rows. Every single one of the columns made use of the int and float data types (J. Chen, 2017). In addition to this, there was the PM 2.5 feature, which served as our aim feature; the other 12 features, on the other hand, served as input features. In this part, you will find descriptions of all of the properties of the dataset. Below is a figure containing basic information about data.

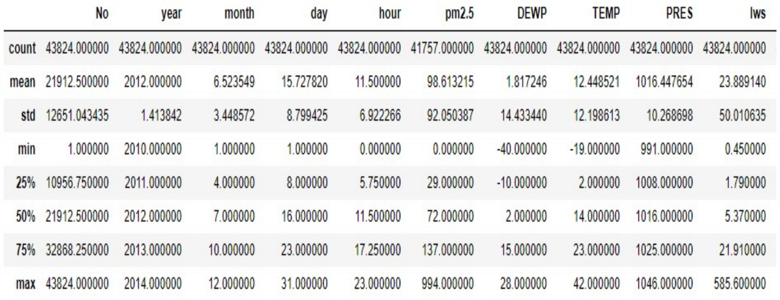


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43824 entries, 0 to 43823
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   No      43824 non-null  int64  
 1   year    43824 non-null  int64  
 2   month   43824 non-null  int64  
 3   day     43824 non-null  int64  
 4   hour    43824 non-null  int64  
 5   pm2.5   41757 non-null  float64
 6   DEWP    43824 non-null  int64  
 7   TEMP    43824 non-null  float64
 8   PRES    43824 non-null  float64
 9   cbwd    43824 non-null  object 
 10  Iws     43824 non-null  float64
 11  Is      43824 non-null  int64  
 12  Ir      43824 non-null  int64  
dtypes: float64(4), int64(8), object(1)
memory usage: 4.3+ MB
```

Figure 2: Basic information about dataset

4.3. Statistical Description of data

I examined the data in terms of their statistical characteristics. These attributes can be found in the table that follows.

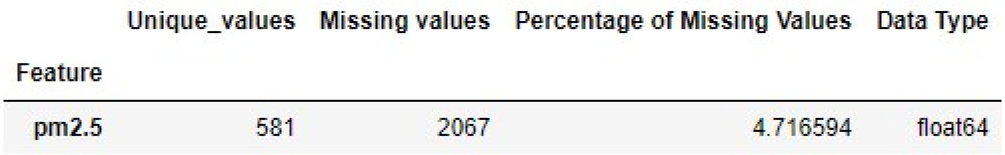


	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	lws
count	43824.000000	43824.000000	43824.000000	43824.000000	43824.000000	41757.000000	43824.000000	43824.000000	43824.000000	43824.000000
mean	21912.500000	2012.000000	6.523549	15.727820	11.500000	98.613215	1.817246	12.448521	1016.447654	23.889140
std	12651.043435	1.413842	3.448572	8.799425	6.922266	92.050387	14.433440	12.198613	10.268698	50.010635
min	1.000000	2010.000000	1.000000	1.000000	0.000000	0.000000	-40.000000	-19.000000	991.000000	0.450000
25%	10956.750000	2011.000000	4.000000	8.000000	5.750000	29.000000	-10.000000	2.000000	1008.000000	1.790000
50%	21912.500000	2012.000000	7.000000	16.000000	11.500000	72.000000	2.000000	14.000000	1016.000000	5.370000
75%	32868.250000	2013.000000	10.000000	23.000000	17.250000	137.000000	15.000000	23.000000	1025.000000	21.910000
max	43824.000000	2014.000000	12.000000	31.000000	23.000000	994.000000	28.000000	42.000000	1046.000000	585.600000

Figure 3: Statistical description of data

4.4. Data Preprocessing

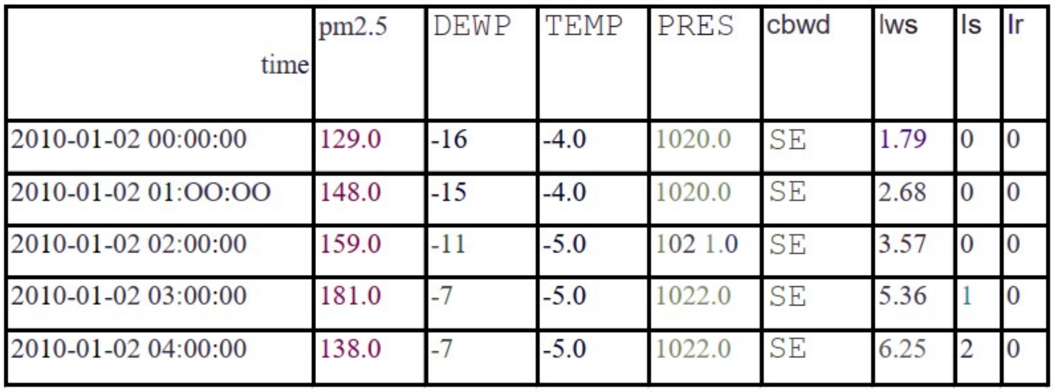
The process of cleaning up data that contains things like missing numbers, outliers, unclear values, and so on is referred to as "data preprocessing." My answer was to look over the data and get rid of everything that was bringing the quality of it down. The first thing that needed to be done was to create a function that would check the dataset for null values, calculate their percentage, and determine the data type of the features that were included in the dataset. After that, I made use of pandas to transform the data into a data frame so that I could hand it off to the primary function of the programme. The application of this function to our dataset led to the discovery of the results that are presented below.



	Unique_values	Missing values	Percentage of Missing Values	Data Type
Feature				
pm2.5	581	2067	4.716594	float64

Figure 4: Missing Values

I simply ignored the first 24 rows, which included no PM2.5 data, and filled in the remaining rows using the panda's library forward filling function. Then merged features for hour, day, month, year to a single feature as well as turned them into a column that Pandas could read (see below). Since we now have the values of every independent date column in a single column, the old date columns have been removed. I made the data frame using the new date column as an index. The image above shows an example dataset with the date column filled in and no null entries.



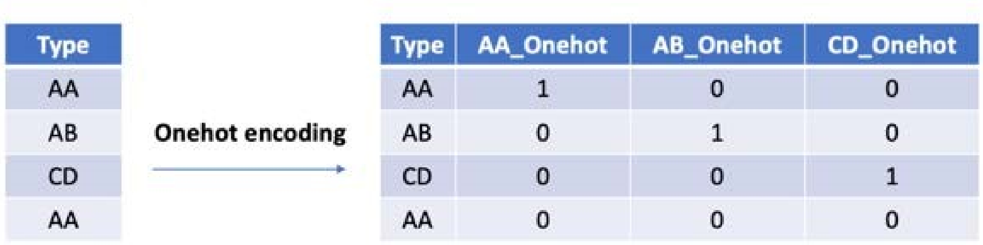
time	pm2.5	DEWP	TEMP	PRES	cbwd	lws	ls	lr
2010-01-02 00:00:00	129.0	-16	-4.0	1020.0	SE	1.79	0	0
2010-01-02 01:OO:OO	148.0	-15	-4.0	1020.0	SE	2.68	0	0
2010-01-02 02:00:00	159.0	-11	-5.0	102 1.0	SE	3.57	0	0
2010-01-02 03:00:00	181.0	-7	-5.0	1022.0	SE	5.36	1	0
2010-01-02 04:00:00	138.0	-7	-5.0	1022.0	SE	6.25	2	0

Figure 5: cleaned dataset

After that I looked at the cbwd column's unique values. The dataset contained the following one-of-akind values.

- NE
- SE
- NW
- CV

Our machine learning algorithms are unable to process the data contained in feature cbwd since it is of the string type. In order to convert this feature from categorical to integer format, I made use of the method known as One Hot Encoding. One hot encoding is a beneficial strategy that helps to aid in accuracy and precision when converting categorical data for use in an algorithm. Each new category column that is produced by using onehot receives either the value 1 or 0 in the binary representation of the value. You can see how each integer value is represented as a binary vector in the image that is included with this description.



Type
AA
AB
CD
AA

Onehot encoding

Type	AA_Onehot	AB_Onehot	CD_Onehot
AA	1	0	0
AB	0	1	0
CD	0	0	1
AA	0	0	0

Figure 6: One hot encoding

As a consequence of this, I made use of the pandas onehot encoder in order to convert our categorical feature, which we designated as cbwd, into a binary representation utilising One hot encoding.

Predicting PM 2.5 index (Mehdipour, 2015) every day, a new data frame had to be constructed, and then the data from each hour had to be added together. The second thing I did was to randomize the data in order to get rid of any patterns I found. After that, which was the penultimate stage, I split the data into two parts: one of which was dependent, and the other of which was independent. At the end of it all, we had datasets in three dimensions.

Following the completion of the procedures that came before it, I fit my data within Standard Scaling approach. When using Standard Scaling, the mean and standard deviation of each feature are transformed to have values of zero and one, respectively, when the scaling method is used. In order to create our model, we made use of the data that we have accumulated up until this point.

4.5. LSTM Model

Due to the time-dependent nature of our data, if the PM 2.5 value is 667 at this very minute, then it will almost definitely still be 667 an hour from now. Both our data and our challenge are time-dependent and time-series based respectively. ANNs struggle with their ability to remember things in the short term. If the series is sufficiently long, they have trouble sending information from earlier time steps to later time steps in the sequence. As a consequence of this, RNNs could skip over essential information at the beginning of the text processing process in order to concentrate on generating predictions.

During back propagation, the vanishing gradient problem manifests itself in recurrent neural networks because of the nature of the network. The utilisation of gradients allows for the ongoing and automatic updating of the weights of neural networks. The problem known as the disappearing gradient occurs when the gradient becomes less significant as one moves further back in time. There is no way to gain any useful information from a gradient whose value has been lowered to an outrageously insignificant portion of the original.

Therefore, in recurrent neural networks, layers that only receive a small update to their gradients are no longer able to learn new information. Those are often the first layers that are put on the structure. As a consequence of this, RNNs that do not contain any learning layers run the risk of losing track of what they have observed over the course of time.

The challenge of having a poor capacity for short-term memory was solved by the invention of LSTM. Gates are internal devices that are able to control the flow of data and can be utilised in a variety of contexts.

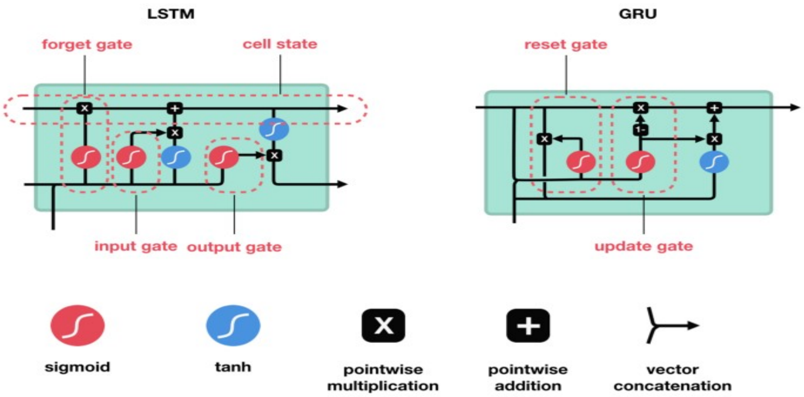


Figure 7: LSTM model

These gates are capable of learning whether data in a sequence is critical to preserve or discard. It can then use that knowledge to create predictions farther down the lengthy chain of sequences. These two networks have been used to create nearly all of the most recent and cutting-edge achievements based on recurrent neural networks. Voice recognition, speech synthesis, and text production all use LSTMs and GRUs. It is possible to utilize them to produce video captions.

5. Experiment

Data was split into two halves with a ratio of 80:20 for training and testing. Batch size and epochs are the hyperparameters of the model that I want to optimise. I adjusted these numbers using Grid Search CV as well. I measured the model using R2 Score, Mean Absolute Error, and Mean Squared Error. Finally, as can be seen in the code, I created a model with the optimal hyperparameter values.

The details of the model are listed below.

- A 32-neuron input layer makes up each of the three LSTM layers.

- Two hidden layers with 32 neurons

The fourth and final Dense layer's output is a single neuron.

Following this procedure, I constructed and trained a model using Adam's optimizer utilising training and testing data as the validation data and mean squared error as the loss. It didn't take the model much longer than ANN. It went through each epoch and gathered knowledge from the information. I did this by plotting the model's training and validation losses using the matplotlib library. This is the plot.

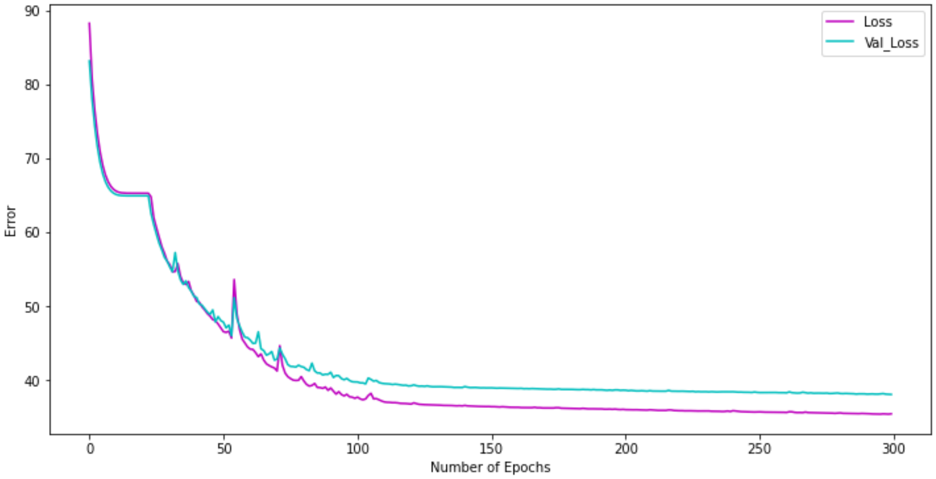


Figure 8: Validation of LSTM on 100 epochs

After that, a number of different evaluation criteria were utilised in order to evaluate the LSTM model. To properly validate the model, it was essential for me to ascertain the quantity of testing data that was required. It lost perhaps in the neighbourhood of 28 percent of its support. After that, I used the model's predictions for testing data to gather the various assessment findings, and the model did a good job considering the amount of time it was given to train. It had an accuracy of approximately 80 percent. And despite this, the performance of the system can be improved by increasing the number of epochs.

The model produces satisfactory results according to our dataset. It is still feasible to enhance its accuracy even further by adding some extra hidden layers, but doing so will make it harder to train and would require a significant amount of effort.

6. Conclusion

As industry and urbanisation have spread rapidly over the world, air pollution has become a worldwide concern. The quality of air in many cities has deteriorated. Air quality testing equipment was not widely accessible in many areas until recently. Smart cities, on the other hand, are becoming increasingly frequent. The idea behind such a city is that everything is mechanised. As a consequence, automating the process of analysing air quality is costly and time-consuming, but predicting air quality based on past data is not. As a consequence, our project is centred on this subject. We intended to develop a technique for estimating the amount of particulate matter 2.5 in the air. The level of air pollution is shown here. As a consequence, we required historical air quality data to build this system. I took the Beijing Air Quality dataset and cleaned and prepared it for the model. Because our issue is time-related, I initially ran an LSTM model on the preprocessed data. As a result, when I evaluated the LSTM model, I was persuaded that it could manage our mission.

It had been going swimmingly. It had an accuracy of almost 80% on test data, although training the model over time may enhance that. This model has four layers in total, although only two of them are visible under the surface. By adding more layers to the model, we can achieve a higher level of precision in our predictions.

7. References

Anon., n.d. [Online].

C. Brokamp, R. J. M. B. R. ,. G. L. a. P. R., 2017. Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches. *Atmospheric Environment,* Volume 151, pp. 1-11.

Delavar, M. R. A. G. G. R. S. Y. R. G. R. N. K. F. a. S. H., 2008. A Novel Method for Improving Air Pollution Prediction Based on Machine Learning Approaches: A Case Study Applied to theCapital City of Tehran. *ISPRS International Journal of Geo,* 2(4), p. 8.

Ding, J. Zhang and W., 2017. Prediction of air pollutants concentration based on an extreme learning machine : The case of Hong Kong. *International Journal of Environmental Research and Public Health,* 14(2), p. 114.

F. Zhang, H. C. a. Z. W., 2014. Fine particles (PM 2.5) at a CAWNET background site in central China : Chemical compositions , seasonal variations and regional pollution events. *Atmospheric Environment,* Volume 86, pp. 193-202.

J. Chen, H. C. ,. Z. W. ,. D. H. a. J. Z. P., 2017. Forecasting smog - related health hazard based on social media and physical sensor. *Information systems,* Volume 64, pp. 281-291.

Johnston, L., 2021. Toxic Chemicals and Pollutants Have an Impact on the Development of Children's Brains. *Journal of Pollution Effects & Control,* p. 305.

Jung, C. -. R. B. -. F. H. a. W. -. T. C., 2005. *Incorporating long - term satellite - based aerosol optical depth, localized land use data, and meteorological variables to estimate ground - level PM 2. 5 concentrations.* Taiwan, s.n.

Liu, H. H. W. X. L. Z. R. M. L. Y. L. a. H. S., 2012. An intelligent hybrid model for air pollutant concentrations forecasting: Case of Beijing in China. *Sustainable Cities and Society,* 471(12), p. 101471.

Mehdipour, V. D. S. S. M. M. f. a. P. S., 2015. *”Comparing different methods for statistical modeling of particulate matter in Tehran, Iran.”.* Tehran, Air Quality, Atmosphere and Health11.

Soh, P.-W. J.-W. C. a. J.-W. H., n.d. Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations. *IEEE Access,* 6(11), pp. 38186-38199.

Soh, P. -. W. K. -. H. C. J. -. W. H. a. H., 2015. *”Spatial - Temporal pattern analysis and prediction of air quality inTaiwan.”.* Taiwan, IEEE.

Sun, X. W. X. a. H. J., 2021. *Spatial - temporal prediction of air quality based on recurrent neural networks.* Hawai, 52th Hawaii International Conference on System Sciences..

Wang, Y. X. H. H. C. L. W. J. B. a. Y. L., 1999. A bayesian downscaler model to estimate daily PM2. 5levels in the conterminous US. *International journal of environmental research and public health,* 9(10), p. 15.

X. Ni, H. H. a. W. D., 2017. Relevance analysis and short - term prediction of PM 2.5 concentrations in Beijing based on multi - source data. *Atmospheric Environment,* Volume 150, pp. 146-161.

X. Xi, Z. W. a. R. X., 2015. A comprehensive evaluation of air pollution prediction improvement by a machine learning method. *In. Proc. IEEE, Tunisia,* pp. 176-181.