Predicting Air Pollutant Concentrations in Beijing

By,
Md Rounak Jahan Raj
rounak.raj@tu-dortmund.de

Submitted to,
Dr. Carsten Burgard
and
Dr. Cornelius Grunwald

Technische Universität Dortmund

July 2025

Abstract

This study explores machine learning techniques to predict hourly concentrations of PM_{10}, SO_2 and NO_2 using the Beijing Multi-Site Air Quality dataset. A Long Short-Term Memory (LSTM) neural network was used as the primary model to capture temporal dependencies, while XGBRegressor served as a feature-based alternative. The data underwent preprocessing, including handling missing values, scaling and encoding cyclical features. LSTM models showed reasonable performance in learning general trends but struggled with sudden fluctuations, especially for SO_2 and NO_2. However, the LSTM model achieved significantly lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) compared to the XGBRegressor, indicating better overall predictive accuracy. In contrast, XGBRegressor produced higher R^2 scores and better captured short-term variations.

Contents

Chapter 1
Introduction

Motivation

Air pollution is one of the most serious environmental challenges of our time. According to the World Health Organization (WHO), it contributes to around 8 million premature deaths every year. Moreover, 99% of the global population is exposed to air that does not meet WHO standards. In an urban city like Beijing, air pollution has been a significant concern due to industrialization and vehicle emissions.

Understanding and forecasting air quality is therefore essential, not only for improving public health policy and urban planning but also for issuing timely alerts to mitigate exposure. Predictive modeling based on environmental and meteorological data can support early interventions and enable informed decision making.

Problem Formulation

This project focuses on analyzing and modeling air quality data for Beijing. The central tasks in this analysis are:

Given historical meteorological and pollutant data, predict future concentrations of key air pollutants—PM_{10}, SO_2 and NO_2 at various urban sites in Beijing.
This predictive task involves:

- Cleaning and preprocessing multivariate time series data
- Exploring trends, correlations and seasonality in pollutant levels
- Building and evaluating a Long Short-Term Memory (LSTM) model, for predicting pollutant concentrations and comparing the results with XGBRegressor model outputs

Chapter 2
Dataset

The dataset used in this analysis was taken from UC Irvine Machine Learning Repository, where it is titled "Beijing Multi-Site Air Quality". This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. This hourly dataset includes six main air pollutants (e.g., $PM_{2.5}$, PM_{10} and SO_2) and six relevant meteorological variables (e.g., temperature and pressure) at multiple sites in Beijing. These air pollutants concentration data were collected from twelve nationally controlled air quality monitoring sites. The meteorological data in each air-quality site are collected from the nearest weather station from the China Meteorological Administration. The dataset spans from March 1st, 2013 to February 28th, 2017 and contains 420,768 rows distributed in seventeen features.

The data for each station was provided as a separate CSV file. After merging all the files into a single dataframe, the first five rows are shown in Figure 2-1.

	year	month	day	hour	PM2.5	PM10	SO2	NO2	CO	O3	TEMP	PRES	DEWP	RAIN	wd	WSPM	station
0	2013	3	1	0	4.0	4.0	4.0	7.0	300.0	77.0	-0.7	1023.0	-18.8	0.0	NNW	4.4	Aotizhongxin
1	2013	3	1	1	8.0	8.0	4.0	7.0	300.0	77.0	-1.1	1023.2	-18.2	0.0	N	4.7	Aotizhongxin
2	2013	3	1	2	7.0	7.0	5.0	10.0	300.0	73.0	-1.1	1023.5	-18.2	0.0	NNW	5.6	Aotizhongxin
3	2013	3	1	3	6.0	6.0	11.0	11.0	300.0	72.0	-1.4	1024.5	-19.4	0.0	NW	3.1	Aotizhongxin
4	2013	3	1	4	3.0	3.0	12.0	12.0	300.0	72.0	-2.0	1025.2	-19.5	0.0	N	2.0	Aotizhongxin

Figure 2-1: Snapshot of the first five entries of the dataset.

Table 2.1 provides a brief description of the dataset. It includes the feature names, their description (with units), number of missing rows for each feature and their data types.

Feature Names	Description	No. of Missing Rows	Dtype
No	Row number	0	int64
year	Year	0	int64
month	Month	0	int64
day	Day	0	int64
hour	Hour	0	int64
PM2.5	$PM_{2.5}$ concentration ($\mu g/m^3$)	8739	float64
PM10	PM_{10} concentration ($\mu g/m^3$)	6449	float64
SO2	SO_2 concentration ($\mu g/m^3$)	9021	float64
NO2	NO_2 concentration ($\mu g/m^3$)	12116	float64
CO	CO concentration ($\mu g/m^3$)	20701	float64
O3	O_3 concentration ($\mu g/m^3$)	13277	float64
TEMP	Temperature (° C)	398	float64
PRES	Pressure (hPa)	393	float64
DEWP	Dew point temperature (° C)	403	float64
RAIN	Precipitation (mm)	390	float64
wd	Wind direction	1822	object
WSPM	Wind sped (m/s)	318	float64
station	Name of the monitoring site	0	object

Table 2.1: Overview of the features in the dataset.

Tables 2-2 and 2-3 give basic statistical information (e.g., mean and median) about the numerical and categorical features respectively.

	year	month	day	hour	PM2.5	PM10	SO2	NO2	CO	O3	TEMP	PRES	DEWP	RAIN	WSPM
count	420768.000000	420768.000000	420768.000000	420768.000000	412029.000000	414319.000000	411747.000000	408652.000000	400067.000000	407491.000000	420370.000000	420375.000000	420365.000000	420378.000000	420450.000000
mean	2014.662560	6.522930	15.729637	11.500000	79.793428	104.602618	15.830835	50.638586	1230.766454	57.372271	13.538976	1010.746982	2.490822	0.064476	1.729711
std	1.177198	3.448707	8.800102	6.922195	80.822391	91.772426	21.650603	35.127912	1160.182716	56.661607	11.436139	10.474055	13.793847	0.821004	1.246386
min	2013.000000	1.000000	1.000000	0.000000	2.000000	2.000000	0.285600	1.026500	100.000000	0.214200	-19.900000	982.400000	-43.400000	0.000000	0.000000
25%	2014.000000	4.000000	8.000000	5.750000	20.000000	36.000000	3.000000	23.000000	500.000000	11.000000	3.100000	1002.300000	-8.900000	0.000000	0.900000
50%	2015.000000	7.000000	16.000000	11.500000	55.000000	82.000000	7.000000	43.000000	900.000000	45.000000	14.500000	1010.400000	3.100000	0.000000	1.400000
75%	2016.000000	10.000000	23.000000	17.250000	111.000000	145.000000	20.000000	71.000000	1500.000000	82.000000	23.300000	1019.000000	15.100000	0.000000	2.200000
max	2017.000000	12.000000	31.000000	23.000000	999.000000	999.000000	500.000000	290.000000	10000.000000	1071.000000	41.600000	1042.800000	29.100000	72.500000	13.200000

Figure 2-2: Snapshot of the basic statistical information of numerical features in the dataframe.

	wd	station
count	418946	420768
unique	16	12
top	NE	Aotizhongxin
freq	43335	35064

Figure 2-3: Snapshot of the basic statistical information of categorical features in the dataframe.

Figure 2-4 provides the distributions of all the numerical features in the form of histograms. These visualizations provide insights into the presence of outliers, skewness of the data and required scaling for each feature.

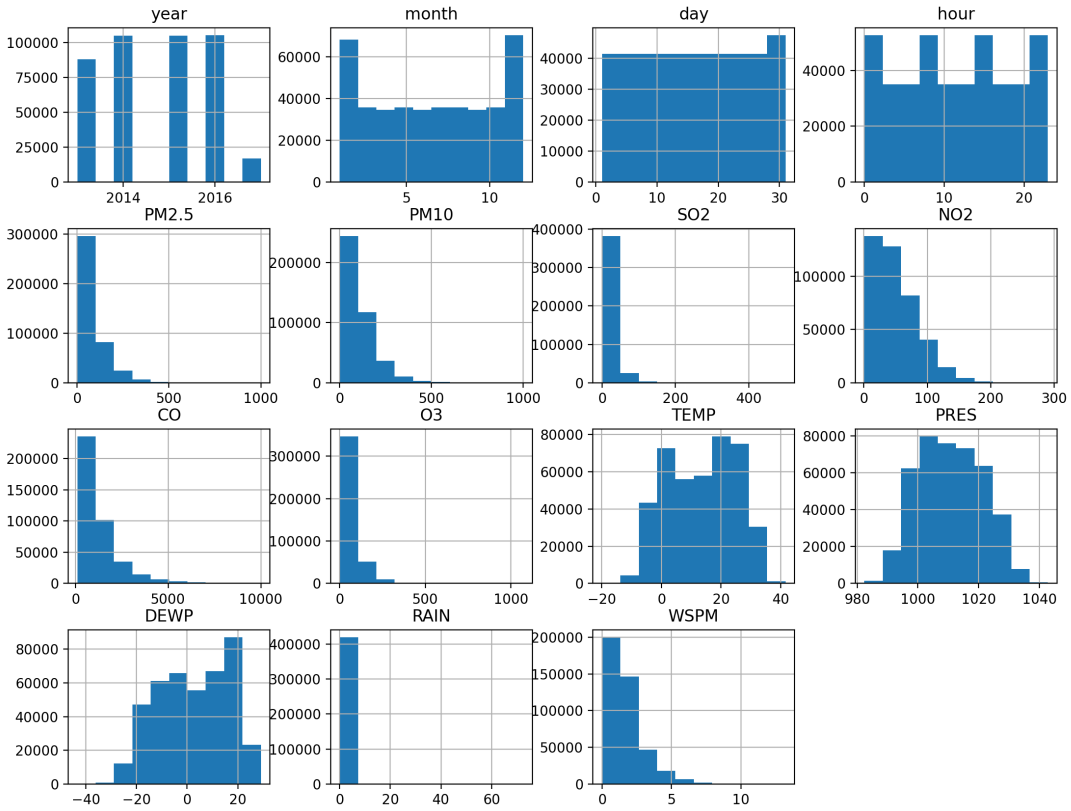


Figure 2-4: Distributions of the numerical features in the dataframe.

The missing values in the dataframe for various pollutants were imputed using rolling average mean and meteorological features were interpolated using datetime. Since, only data of five years are in the dataframe, "year" column doesn't contain much informa-

tion and hence removed during preprocesssing. Sine and cosine components were obtained for the "month", "day" and "hour" as they are cyclic features. Highly skewed features ("PM2.5", "PM10", "SO2", "NO2", "CO", "O3", "RAIN" and "WSPM" were pass through a pipeline where they first transformed using np.log1p and then with RobustScaler. The rest fo the features were scaled using StandardScaler since their distributions are normal. Moreover, as LSTM works best for bounded targets. Hence, the target for each pollution prediction were passed through another pipeline, which used MinMaxScaler after np.log1p.

Chapter 3
Solution Approach

The primary model used in the analysis is LSTM. It is a type of recurrent neural network (RNN) aimed at mitigating the vanishing gradient problem commonly encountered by traditional RNNs [1]. LSTM networks are particularly effective for capturing temporal dependencies and non-linear interactions between target and features. The link for the

3.1 LSTM Model

The dataset contains hourly measurements of pollutants and meteorological variables. LSTMs are built to learn such time-dependent trends by retaining memory of previous inputs over long sequences. Moreover, pollution levels at a given hour strongly depend on previous hours' conditions (e.g., accumulation during the night, rush hour spikes, weather shifts). LSTM's gated architecture allows it to retain important information over long time lags, which is crucial for accurate pollution prediction. Table 3.1 shows the best model architectures for predicting PM_{10}, SO_2 and NO_2.

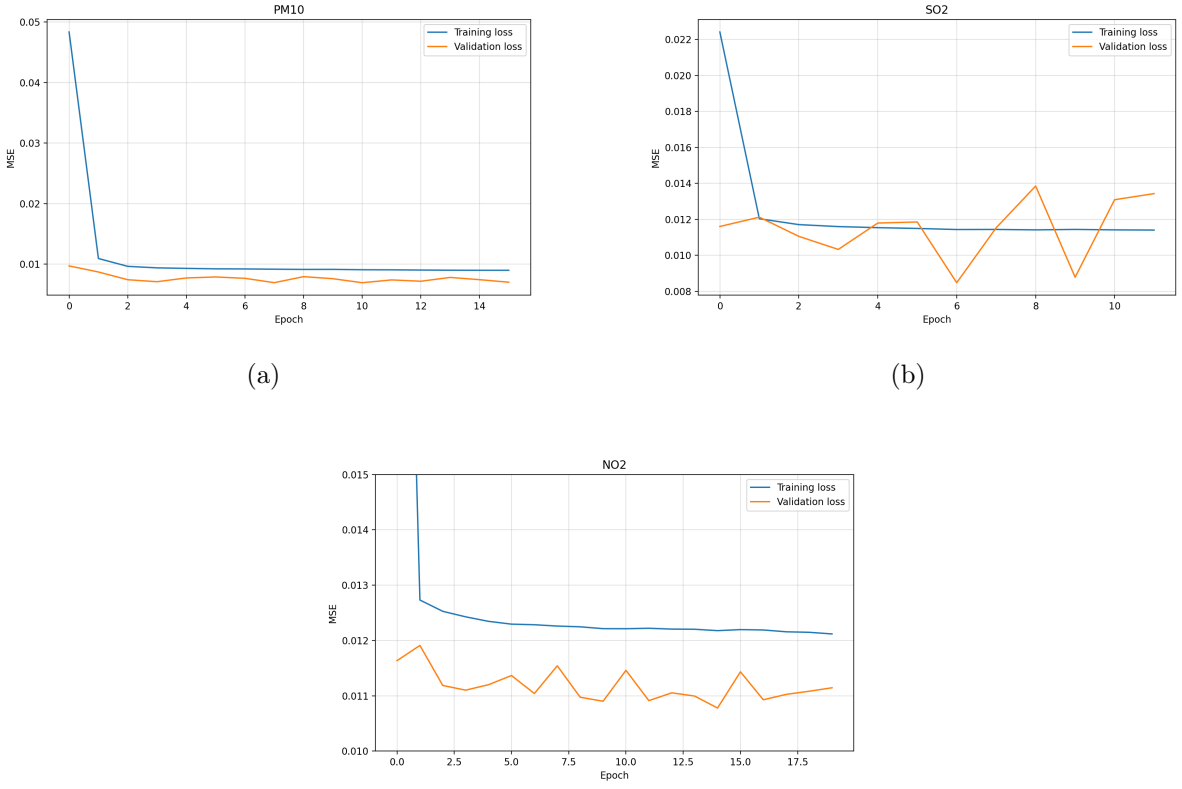
Layers (type)	**PM_{10} and NO_2**		**SO_2**	
	Output Shape	**Param #**	**Output Shape**	**Param #**
lstm (LSTM)	(None, 24, 64)	30,464	(None, 24, 32)	11,136
dropout (Dropout)	(None, 24, 64)	0	(None, 24, 32)	0
lstm_1 (LSTM)	(None, 16)	5,184	(None, 16)	3,136
dropout_1 (Dropout)	(None, 16)	0	(None, 16)	0
dense (Dense)	(None, 1)	17	(None, 1)	17

Table 3.1: Model architectures for predicting PM_{10}, SO_2 and NO_2.

The hyperparameters for the best models were obtained using RandomizedSearchCV by minimizing the mean squared error and shown in Table 3.2.

Hyperparameters	**PM_{10}**	**SO_2**	**NO_2**
units1	64	32	64
units2	16	16	32
dropout_rate	0.3	0.2	0.2
l2_rate	0.001	0.01	0.001
learning_rate	0.0001	0.001	0.001
batch_size	32	32	32
epochs	25	25	15
optimizer	rmsprop	rmsprop	rmsprop

Table 3.2: Hyperparameters for best LSTM models obtained using RandomizedSearchCV.



(a) (b)

(c)

Figure 3-1: Training and validation loss curves for PM_{10}, SO_2 and NO_2.

The LSTM models were trained using sequences of 24 previous hours to predict the pollutant concentration for the next hour. Training and validation loss plots as shown in Figure 3-1 confirmed stable training with no overfitting, aided by dropout layers and L2

regularization. The use of dropout layers, which only work on training dataset and not on validation dataset, in the models have resulted in lower validation losses compared to training losses.

3.2 XGBRegressor Model

The XGBRegressor (Extreme Gradient Boosting) was chosen as an alternative solution to LSTM. Although the data has a temporal component, the pollutant concentration levels are also strongly influenced by meteorological and spatial features (temperature, pressure, wind, station ID, etc.), making the dataset suitable for tabular models like XGBRegressor. The dataset used for the analysis contained noisy and missing values. XGBRegressor can handle these more robustly than neural networks, which require careful preprocessing. Additionally, it is very easy and fast to implement because it does not require the input features to have values within a boundary. The hyperparameters for the best models for the alternatie approach were found using GridSearchCV and presented in the Table 3.3.

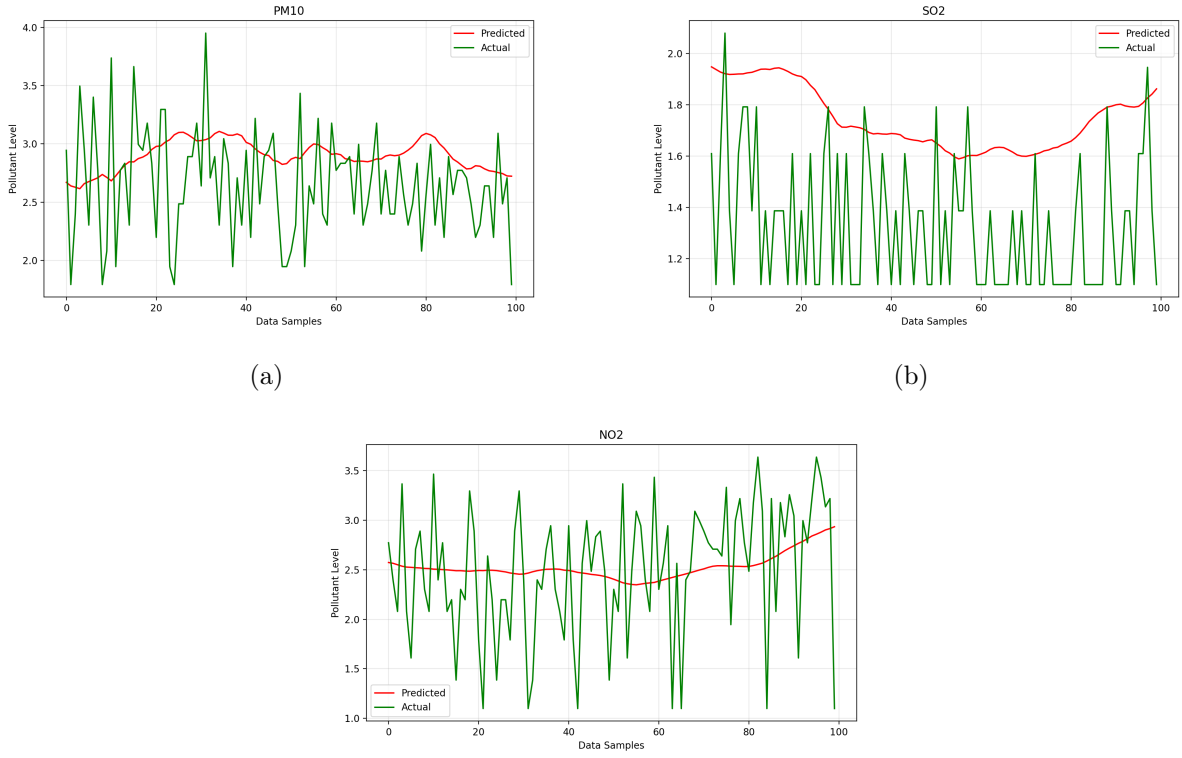
Hyperparameters	**PM_{10}**	**SO_2**	**NO_2**
n_estimators	200	300	300
learning_rate	0.1	0.1	0.1
max_depth	8	8	8
min_child_weight	1	1	1

Table 3.3: Hyperparameters for XGBRegressor model obtained using GridSearchCV.

Chapter 4
Results

4.1 LSTM Model

Figure 4-1 shows the predicted versus actual values for the first hundred samples of the test set.



(a)

(b)

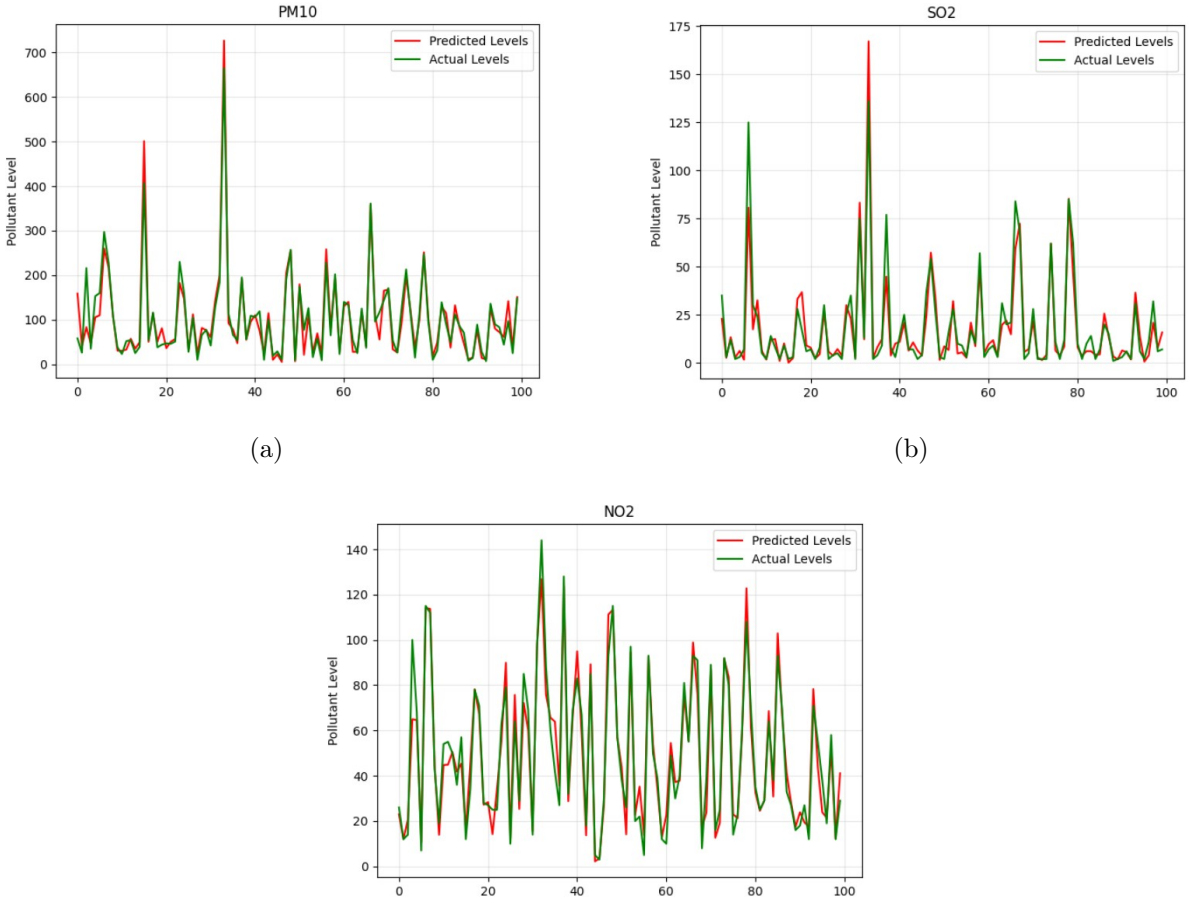
(c)

Figure 4-1: Predicted versus actual pollutant concentration level for the fist hundred test samples using LSTM.

Visually, the predictions are smooth and follow the general trend of the ground truth. However, due to the smoothing nature of LSTM and regularization, the models sometimes underpredict sharp peaks and sudden drops in pollutant levels resulting low R^2 score.

4.2 XGBRegressor Model

Figure 4-2 shows predicted versus actual values for the first hundred test samples. Visually, the XGBRegressor captures the dynamic fluctuations much more sharply than the LSTM, responding better to spikes and dips in pollution levels.



(a) (b)

(c)

Figure 4-2: Actual versus predicted pollutant concentration level for the fist hundred test samples using XGBRegressor.

4.3 Comparative Analysis

The performance of the primary model (LSTM) is compared with the alternative approach (XGBRegressor). The evaluation is based on three common regression metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score. Table 4.1 shows evaluation results for both the primary and alternatie approaches.

Pollutant	**LSTM**			**XGBRegressor**		
	MAE	**RMSE**	**R^2 Score**	**MAE**	**RMSE**	**R^2 Score**
PM_{10}	0.36	0.49	0.75	15.67	3.97	0.91
SO_2	0.46	0.61	0.56	3.99	2.00	0.89
NO_2	0.44	0.55	0.57	6.77	2.6	0.93

Table 4.1: Evaluation results for PM_{10}, SO_2 and NO_2 obtained from both LSTM and XGBRegressor.

The R^2 scores show that LSTM captures a significant portion of the variance in the data, though performance varies across pollutants. PM_{10} is predicted relatively well, while SO_2 and NO_2 predictions show reduced accuracy, possibly due to more irregular temporal behavior or lower signal-to-noise ratio.

For the XGBRegressor model, despite the higher absolute error values (MAE and RMSE) due to unscaled targets, the R^2 scores are significantly higher than those of the LSTM models, indicating stronger correlation with true values and better ability to capture variance in the data.

Chapter 5
Summary

This study explored the application of machine learning models to predict air pollutant concentrations in Beijing using the publicly available Beijing Multi-Site Air Quality dataset. The pollutants of interest—PM_{10}, SO_2 and NO_2 —were predicted using two approaches: a Long Short-Term Memory (LSTM) neural network and the XGBRegressor algorithm.
The LSTM model, serving as the primary approach, was selected for its ability to capture long-range temporal dependencies inherent in time series data. The input sequences were formed from 24-hour windows of previous data to predict the concentration level for the next hour. Despite the model's ability to follow general trends in pollutant levels, it showed limitations in capturing sharp peaks and sudden drops—likely due to the smoothing effect of regularization and dropout layers. This behavior was reflected in the relatively modest R^2 scores, especially for SO_2 and NO_2.

In contrast, the XGBRegressor, used as an alternative approach, produced better performance in terms of predictive accuracy across all three pollutants. It was more responsive to fluctuations in the data and achieved significantly higher $\mathbf{R^2}$ scores. This suggests that while LSTM is well-suited for sequence modeling, its performance may be constrained in cases where the temporal signal is noisy or subtle, and where tree-based models can exploit rich tabular and spatial-meteorological features more effectively.
However, both models have limitations. The LSTM models require significant data preprocessing and hyperparameter tuning, and may underperform if not provided with enough relevant features or sufficiently long training sequences. On the other hand, XGBoost treats time as a tabular feature, lacking true sequence awareness. As such, it may fail to generalize to long-term dependencies or delayed effects unless those are carefully engineered into the feature set.

Limitations and Future Improvements

- Data Coverage: The dataset spans only five years and contains missing values and seasonal gaps, which may reduce the robustness of the LSTM model on unseen

scenarios.

- Feature Scope: Future work could include additional features such as traffic data, industrial activity levels, or satellite-based pollution indices to enrich the input space.
- Temporal Resolution: Testing with different sequence lengths (e.g., 48 or 72 hours) might help the LSTM model learn longer-term dependencies more effectively.
- Model Architectures: Incorporating advanced architectures such as Bidirectional LSTM, GRU, or attention mechanisms may help overcome the limitations of traditional LSTM.
- Ensemble Models: Combining temporal models (LSTM) and tabular models (XGBoost) in an ensemble could capture both sequence dependencies and static feature importance.

Conclusion

Overall, this study demonstrates that while LSTM provides a theoretically sound approach for time-series prediction, models like XGBRegressor may offer better performance in real-world, noisy environmental data scenarios. The comparative analysis suggests that the choice of model should depend not only on the temporal nature of the task but also on the quality and structure of the data. With further improvements and deeper feature engineering, machine learning models can play a crucial role in air quality forecasting and public health interventions.

Literatures

[1] "Long short-term memory". https://en.wikipedia.org/wiki/Long_short-term_memory.

Appendix A: Code and Resources

The code used for this project is available on Google Colab and can be accessed via the following link: Google Colab Notebook