Air Quality Forecasting Using LSTM Models: A Comprehensive Report

1. Introduction

Air pollution is a critical environmental and public health issue globally, particularly in urban centers like Beijing. Among various pollutants, PM2.5 (particulate matter with a diameter of 2.5 micrometers or less) poses the greatest health risk as it can penetrate deep into the lungs and even enter the bloodstream. Prolonged exposure to elevated PM2.5 levels has been linked to respiratory and cardiovascular diseases, increased hospital admissions, and premature mortality [1][2].

Forecasting PM2.5 concentrations accurately is essential for timely public health interventions, policy-making, and urban planning. Machine learning techniques, especially those designed to handle sequential data, offer promising solutions. In this project, Long Short-Term Memory (LSTM) architectures were utilized to predict PM2.5 levels based on historical air quality and meteorological data from Beijing. The goal was to minimize the Root Mean Squared Error (RMSE) to under 4000 on a Kaggle leaderboard.

2. Data Exploration and Preprocessing

The dataset contained hourly PM2.5 and meteorological readings over three years (2010–2013). Exploratory Data Analysis (EDA) revealed missing values, particularly in the PM2.5 column. Due to the presence of long sequences of missing values (up to 155 consecutive entries), linear interpolation was chosen to handle those missing values, followed by forward and backward filling. This method preserved trends and was computationally efficient.

Also,

- Converted timestamps to datetime format and set them as indices.
- Created visualizations (boxplots, time series, rolling averages) to gain new insights from the data.
- Observed seasonality and extreme pollution events.
- Identified outliers and correlations with weather variables (e.g., temperature, dew point).

Features were scaled using MinMaxScaler, and the input was reshaped to fit LSTM input requirements (samples, timesteps, features).

3. Model Design and Architecture

Both standard and bidirectional LSTM models were developed and iterated through a number of varying optimizers, learning rate, early stopping use and many other parameters using a modular function that allowed for experimentation with:

- Number of layers and units (e.g., [128, 64], [256, 128, 64])
- Optimizers: Adam, RMSprop, SGD
- Learning rates (e.g., 0.0003, 0.0002, 0.01)
- Dropout rates (0.0–0.3)
- Activation functions: tanh, relu
- Bidirectional wrapping

Early stopping was implemented to prevent overfitting. Models were evaluated based on training RMSE.

4. Findings

Model	Layers	Units	Optimizer	LR	Dropout	Bidirectional	Key Result (Training MSE)
Model 1	1	[32]	Adam	-	0.0	No	6210.8
Model 2	2	[128, 64]	RMSprop	0.0008	0.3	No	5432.2
Model 3	2	[128, 64]	Adam	0.0003	0.2	No	5840.6
Model 4	2	[128, 64]	RMSprop	0.0003	0.15	No	8101
Model 5	3	[256, 128, 64]	RMSprop	0.0002	0.1	No	8720
Model 6	2	[128, 64]	Adam	0.0003	0.2	Yes	5591.5
Model 7	3	[256, 128, 64]	RMSprop	0.0002	0.15	Yes	8377
Model 8	3	[128, 64, 32]	RMSprop	0.0005	0.3	Yes	5329.9
Model 9	2	[128, 64]	SGD	0.01	0.25	Yes	5563.8
Model 10	1	[256]	Adam	0.0002	0.1	Yes	7395.1

The table above summarizes architecture, hyperparameters, optimizer choices, bidirectionality, and training RMSE. The best models utilized deeper LSTMs (2-3 layers), RMSprop with small learning rates, and bidirectional configurations.

Best performance was achieved by **Model 8** with a training RMSE of approximately ~7100 using:

- 3 layers: [128, 64, 32]
- Optimizer: RMSprop
- Learning rate: 0.0005
- Dropout: 0.3
- Bidirectional LSTM

5. Results and Discussion

There is no straight variation about the parameters. Some basic LSTMS performed better than Bidirectional LSTMs when the expectations before training was that all bidirectional LSTMS should perform better than basic one, which was not the case.

Some key findings:

- Adam and RMSprop (with learning rates ~0.0002–0.0005) performed better than SGD, which required careful tuning and more epochs to converge.

- Lower learning rates (e.g., 0.0002) helped stabilize training, while higher rates (e.g., 0.01 with SGD) led to erratic loss curves.

- Models with multiple LSTM layers (e.g., 256-128-64 units) and dropout (0.1–0.3) showed better generalization than shallow networks generally. This goes without saying that early stopping played a crucial part in enhancing a model's performance

6. Data Challenges Impacted Performance

- The skewed distribution of PM2.5 values (with extreme outliers) likely inflated MSE

7. References

[1] World Health Organization. "Air pollution." https://www.who.int/health-topics/air-pollution#tab=tab_1

[2] Dominici, F., et al. "Fine particulate air pollution and hospital admission for cardiovascular and respiratory diseases." JAMA 295.10 (2006): 1127-1134.

[3] Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780.

[4] Srivastava, N., et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research, 15(2014), 1929-1958.

[5] Brownlee, J. "How to Reshape Input for LSTM Networks in Keras." Machine Learning Mastery. https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/