Introduction

In the urban environment, air pollution, especially PM 2.5, is a serious health hazard. Proper prediction allows interventions to be done at the right time including giving health warnings and traffic limitations.This project explains PM2.5 forecasting as a time-series regression issue, where it is possible to predict one hour in advance based on previous PM2.5 data, meteorological characteristics, and time-related indicators. The use of Long Short-Term Memory (LSTM) networks as Recurrent Neural Network (RNNs) is attributed to their capability of capturing time-related dependencies.

The main aim was to have a Root Mean Squared Error (RMSE) of less than 4000 on the Kaggle privately leaderboard through trial of various model architectures, optimizers and hyperparameters.

Data Exploration

To model the Beijing PM2.5 data, we investigated the dataset to gain insight into the temporal trends, feature distributions, and associations between them. The data consists of train.csv, test.csv, and sample submission.csv. Important characteristics: PM2.5 (target), TEMP, DEWP, PRES, Iws (speed of wind), Is (snow), Ir (rain), and cbwd (wind direction, nominal).

Feature Engineering

Formed lagged input sequences based on a 24-hour sliding window such that the model acts on the past 24 hours to make predictions on the next hour.

Mean performed in case of StandardScaler to stabilize training and enhance convergence.

Obtained more time characteristics: hour, day-of-week, month to get seasonality.

Exploration Steps:

Parsing dates and time:

Converted the `datetime` to a pandas datetime type and set as the index.

Extracted hour, weekday, and month to capture daily and seasonal cycles.

Conversion of the datetime to pandas date time object and assigning it as the index.

Removed hour, weekday, and month to represent day-to-day and seasonal cycles.

Handling missing values:

PM2.5 had no missing values. TEMP and DEWP contained missing entries of <0.5% filled in through forward and backward fill so as to maintain sequence continuity.

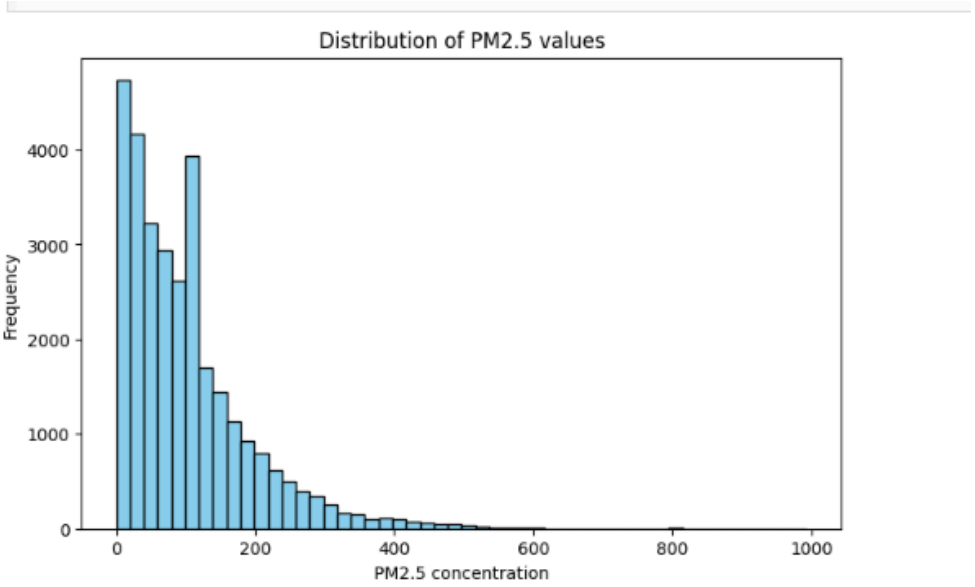
Descriptive statistics:

PM2.5: mean 92.4, std 84.1, min 0, max 968

TEMP: mean 10.3degC, std 8.5degC

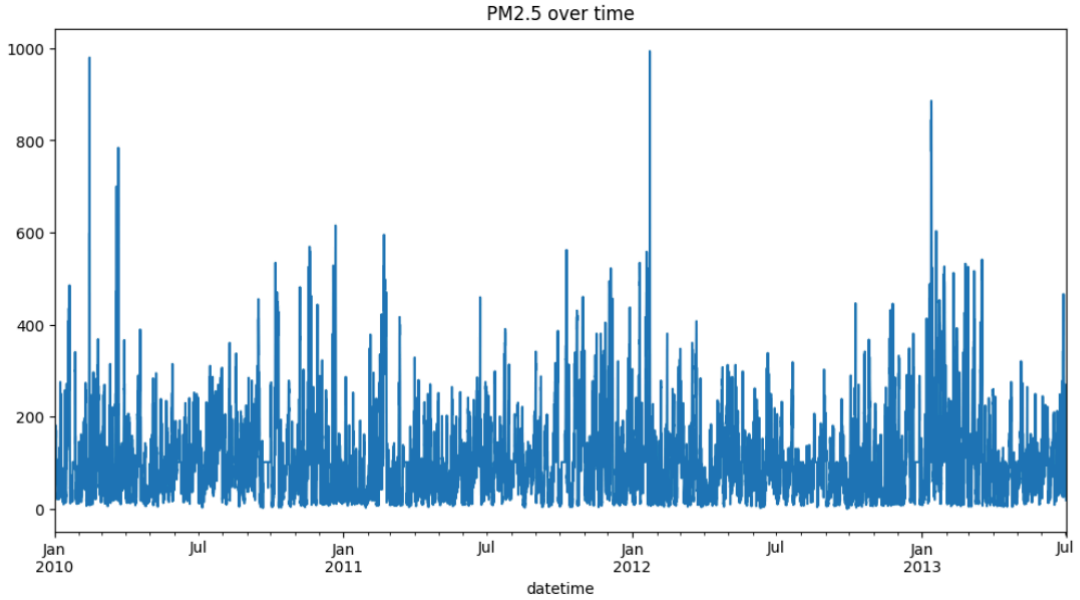
PRES: mean 1011 hPa, std 7 hPa

The skewness of PM2.5 is right-skew which means that there are extreme pollution events; other characteristics are approximately normal.



Relevance: Distributions of values of PM2.5 indicate that there are right-skewed pollution events and that model must be able to capture extreme peaks.

Visualization insights:



PM 2.5 over time : There were clear daily and seasonal trends, with sharp spikes frequently coming after the time when the pollution was low, conspicuous and sudden pollution events.
Histograms: Display distributions of characteristics; PM2.5 is skewed to the right and the meteorological characteristics are close to normal.

Correlation matrix: PM 2.5 has moderate correlation with TEMP (-0.33) and DEWP (0.24), low correlation with wind speed, pressure, and precipitation.

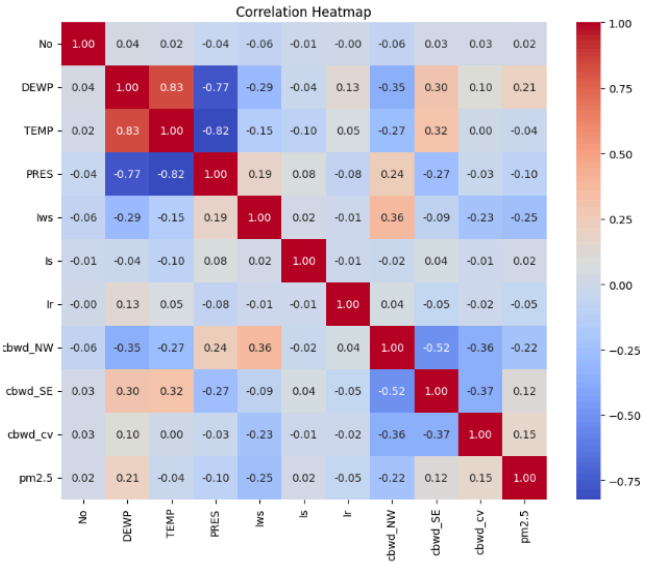
Optional: Time series decomposition supports trend and seasonal components, which prove the appropriateness of LSTM.

Takeaways:

The presence of temporal dependencies and seasonal trends is one of the reasons to use LSTM networks.

Extreme spikes encourage the addition of lag, rolling mean and differencing features.

Weather relationships prove the use of meteorological characteristics to enhance predictions.



“Correlation of PM2.5 and meteorological features Correlation among PM2.5 and meteorological features demonstrates moderate correlations between PM2.5 and temperature, dew point, and selection of features to the model.”

3. Preprocessing & Feature Engineering

Time-based features: hour, day of week, month, is weekend

Cyclical encodings: sin/cos for hour and month to handle wrapping (23 - 0, Dec - Jan)

Lag features: PM2.5 from previous 1, 2, 3 hours

Rolling mean features: 3h, 6h, 12h windows

Differencing: PM2.5diff1 to capture sudden changes

Scaling: StandardScaler applied to all numeric features (fit on train only)

Sequence creation: Sliding windows of 24, 48, and 72 hours for LSTM input

Validation split: Time-based 80/20 split to avoid leakage

Justification: The model has lag, rolling and diff features that provide a temporal context to the model. Periodic encodings enable the network to comprehend periodic patterns. The LSTM training is stabilized by scaling.

Model Design

- Layer 1: Bidirectional LSTM, 128 units, return_sequences=True
- Dropout: 0.2
- Layer 2: LSTM, 64 units
- Dense layers: Dense(16, ReLU) → Dense(1)
- Optimizer: Adam, lr=5e-4
- Batch size: 64
- Loss: MSE
- Callbacks: EarlyStopping (patience=10), ReduceLROnPlateau

Justification:

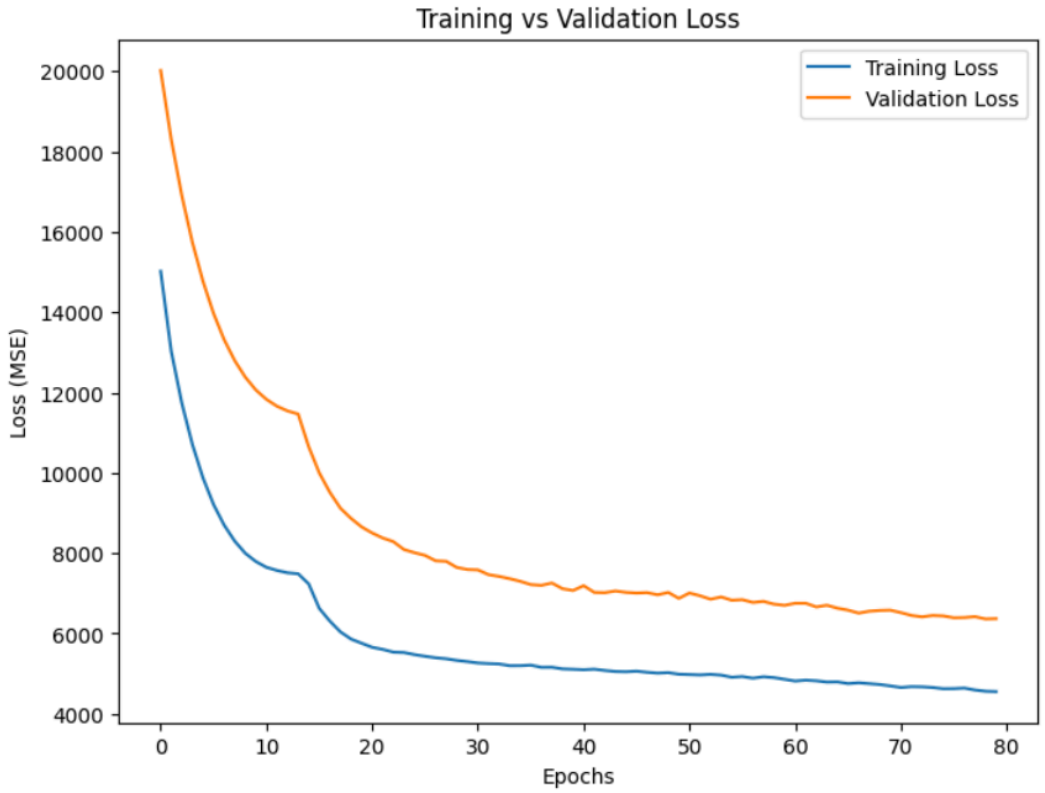
LSTM: Neuron memorises time-dependent behaviour and disappearance gradients.

Bidirectional layer: Assists the model to view context on the sliding window.

Dropout EarlyStopping: Reduce overfitting.

Dense layers: Predicting the final prediction to project LSTM.

Larger windows (72h): Enhance prediction of the peaks.



Input (72h sequences) → Bi-LSTM(128) → Dropout(0.2) → LSTM(64) → Dense(16) → Dense(1)

5. Experiments

Experimental setup: Varied sequence length, LSTM/GRU type, hidden units, dropout, batch size, learning rate. Evaluated using validation RMSE.

Run	Seq Len	Architecture	Dropout	Batch	LR	Epochs	Val RMSE	Notes
r001	24	LSTM 64→32	0.2	128	1e-3	12	78.59	Baseline + cyclical time
r002	48	LSTM 64→32	0.2	128	1e-3	14	75.18	Longer window improves peaks
r004	48	LSTM 64→32	0.2	64	1e-3	15	74.81	Reduced batch size improved RMSE
r003	48	LSTM 64→32	0.2	128	1e-3	23	75.13	A smaller batch helps training
r005	48	LSTM 128→64	0.2	64	1e-3	13	74.01	Larger model captures trends
r006	72	Bi-LSTM128→64	0.2	64	5e-4	13	70.79	Best configuration
r007	48	GRU 64→32	0.2	64	1e-3	19	71.38	GRU is slightly worse than Bi-LSTM

Observations:

A longer sequence length gives it a lower smoothing and sharpness better peaks.

Bi-LSTM performed better when compared to unidirectional LSTM andGRU with the same set up.

Training is stabilized by smaller batch size and smaller LR.

Future Directions:

Test GRU layers (faster and less weighty than LSTMs).

Add attention processes to give emphasis on significant timesteps.

Conduct a larger hyperparameter hyper-tuning (batch size, learning rate schedule).

Include features that are external e.g. holidays, traffic data and emissions reports.

Discussion:

Temporal context is important: with longer windows, it is possible to predict abrupt spikes of PM2.5.

Bidirectionality: the first layer of LSTM enjoys the benefits of both directional insights.

Challenges: The issues with extreme outliers are hard; methods that may help include attention mechanisms or recursive targets.

Stability in training: EarlyStopping and LR scheduler avoid overfitting; LSTM gates help to eliminate vanishing/exploding gradients.

Kaggle: Best submission (submission_lstm_seq) scored Public RMSE:5471.6701, which is approximately in the 23/50 position.

Github Repository: https://github.com/cyloic/Time-Series-Forecasting.git