MULTI-STEP AIR QUALITY FORECASTING

CS365 DEEP LEARNING
PROJECT REPORT

Atul Kumar

2101AI08

Department of Computer Science Engineering

Indian Institute of Technology Patna
atul_2101ai08@iitp.ac.in

November 2023

1 Introduction

Currently, many developing countries are still suffering from air pollution problems. Air pollution can cause serious health issues like respiratory disease, cardiovascular disease, and reduced lung function. Therefore, controlling air pollution has attracted widespread public attention. To reduce the damages made by air pollutants, the concentration prediction method concerning intelligent data analysis approaches has received increasing attention in past decades.

$PM_{2.5}$ (particulate matter with an aerodynamic diameter of less than 2.5 μm) is one of the most dangerous air pollutants because it causes the most adverse effects on public health and can spread through the air. Thus, the $PM_{2.5}$ concentration problem is a global issue that crosses geographical boundaries.

Traditional methods like ARMA cannot process a large amount of multidimensional non-linear data, and analysis of the correlation features between $PM_{2.}5$ concentration and other climate variables is complex for these methods. In this project, our ultimate goal is to construct a PM2.5 concentration prediction model with improved accuracy over the currently employed prediction models using real-world meteorological data.

2 Project Idea

The proposed model adopts a comprehensive approach to time series prediction for $PM_{2.5}$ concentration. It decomposes the input time series features into trend and seasonal components by applying 1D average pooling. These components are then individually processed using two LSTM layers or two LSTM auto-encoders, each with 64 units, to capture temporal dependencies and patterns within the data. The hidden states of these LSTM layers are expanded over a specified prediction horizon. Finally, time-distributed dense layers are applied to these decoder inputs, yielding separate predictions for the trend and seasonal components.

The model's architecture is designed to integrate these predictions, offering a holistic forecast for PM2.5 concentration. This approach allows the model to leverage LSTM networks for effective temporal feature extraction. It provides a structured methodology for handling complex time series data in the context of air quality prediction. Furthermore, the model's architecture may offer resilience in noisy data. Incorporating an initial step for time series decomposition, separating the trend and seasonal components, acts as noise reduction.

3 Proposed Methodology

The proposed pipeline consists of four components: data preprocessing, time-series decomposition, temporal feature extraction, and prediction layer.

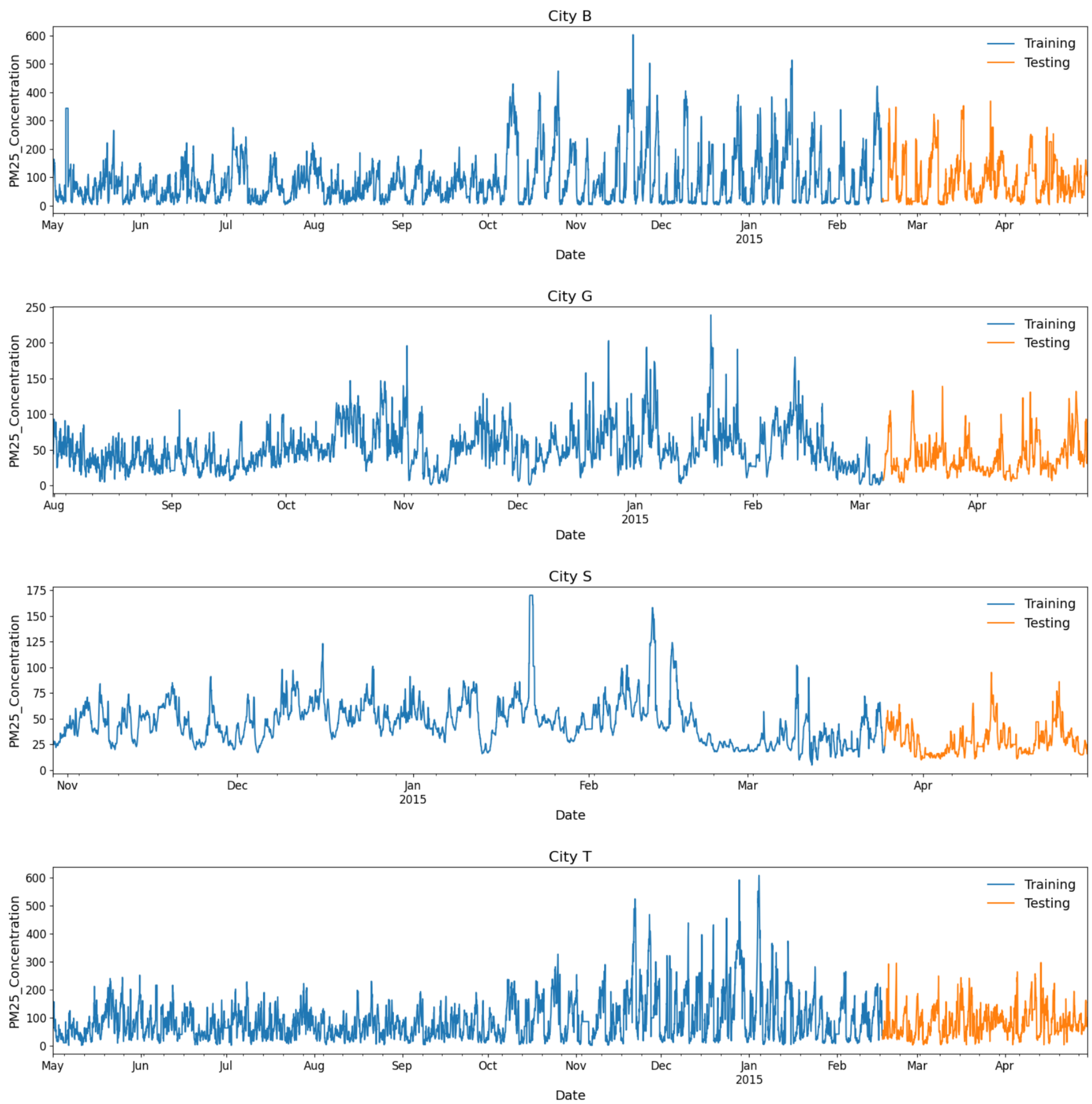


Figure 1: Visualising Time Series Plots

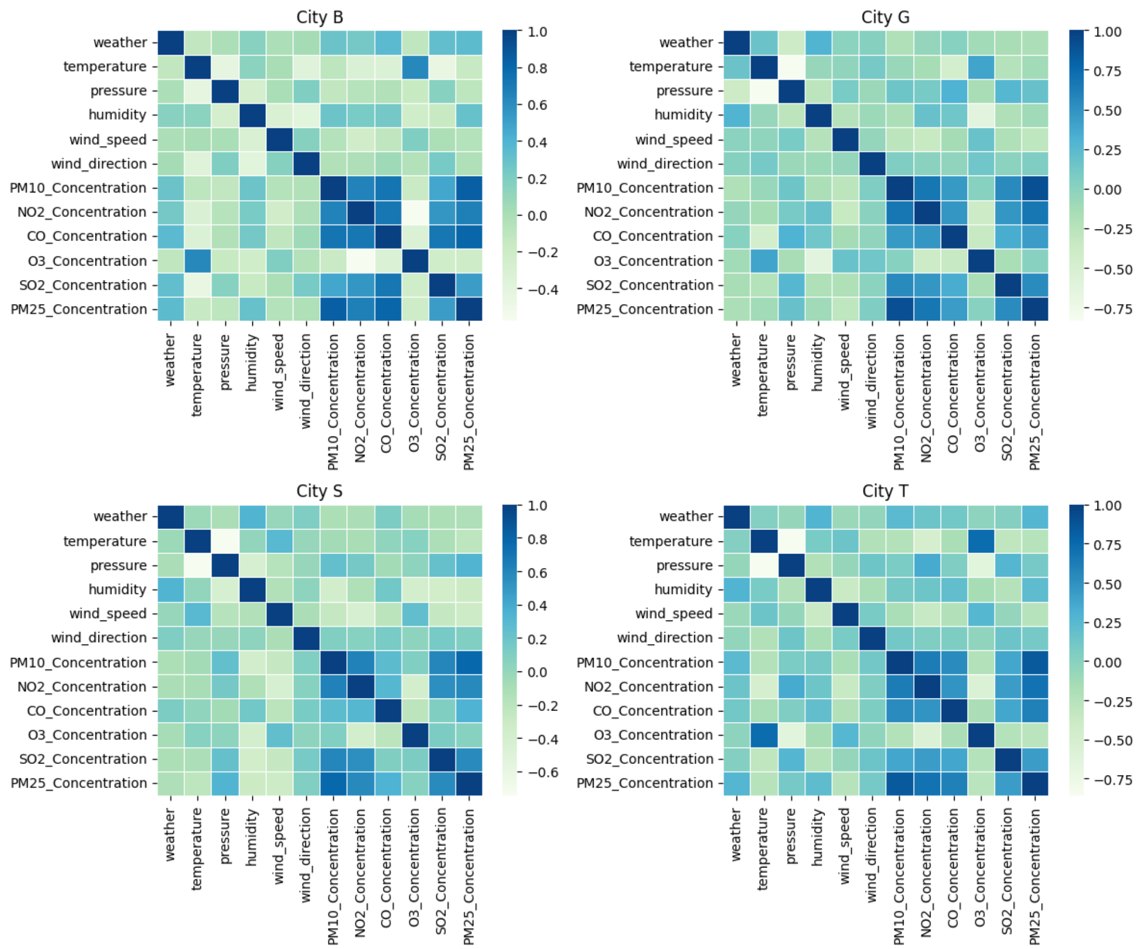


Figure 2: Correlation between features

3.1 Data Preprocessing

We have used the air quality dataset of cities B, G, S, and T, which contains hourly measurements of $PM_{2.5}$ concentrations.

3.1.1 Data Normalization

The dataset is normalized using the minimum-maximum normalization technique, which can be computed as

$$X_{scaled} = \frac{X - min}{max - min} \tag{1}$$

where max and min represent the maximum and minimum data values, respectively. The method can preserve the relationships that exist in the original data.

3.1.2 Data Sliding Window

This operation mainly creates time series data by the predefined sliding window size and step for the original data. In other words, this operation generates the predicted data for the next moment using historical data with a given interval. Given any time series data with length N, such as $\{1, 2, 3, 4, 5, ..., N-1, N\}$, when the sliding window size is set to L, and the sliding step is 1, the $N-L$ data sets with length $L+1$ are formed. The first L data of each set is regarded as training data, and the value of the number $L+1$, ... is the target value.

3.2 Time-Series Decomposition

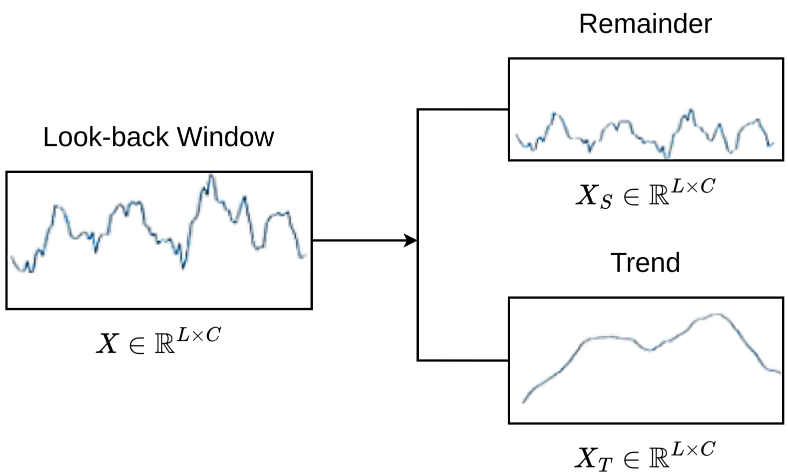


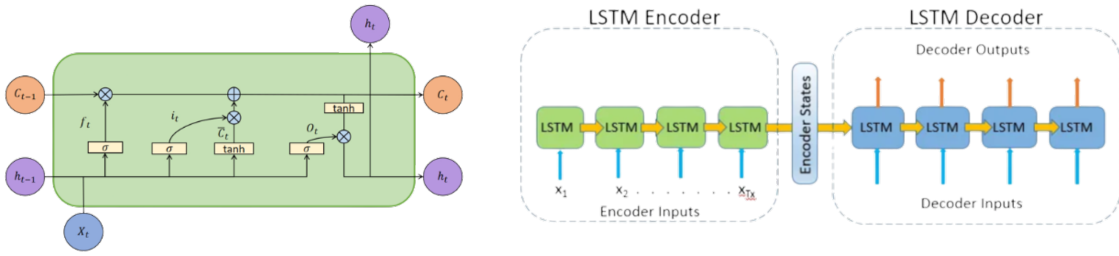
Figure 3: Time Series Decomposition

We have used a moving average kernel on the input sequence to extract underlying components of the time series, namely, trend and seasonality, as shown in Figure 3. The seasonal component is the difference between the original sequence and the trend component.

3.3 Temporal Feature Extraction

The LSTM neural network is well-suited for processing, classification, and prediction tasks on time series data. LSTM is capable of learning long-term dependencies without retaining redundant context information. We have used two LSTMs - one for the seasonal component and the other for the trend component.

In other architecture, we used the LSTM auto-encoder network for seasonal and trend components. This architecture comprises two models: one for reading the input sequence



(a) LSTM Block

(b) Encoder-Decoder LSTM

Figure 4: Two Architectures for modeling the trend and seasonality components

and encoding it into a fixed-length vector and a second for decoding the fixed-length vector and outputting the predicted sequence.

3.4 Prediction Layer

The extracted features and learned temporal dependencies are combined to make air quality predictions for future time steps. The prediction layer is a simple feed-forward neural network that learns to map the extracted features to the predicted air quality values.

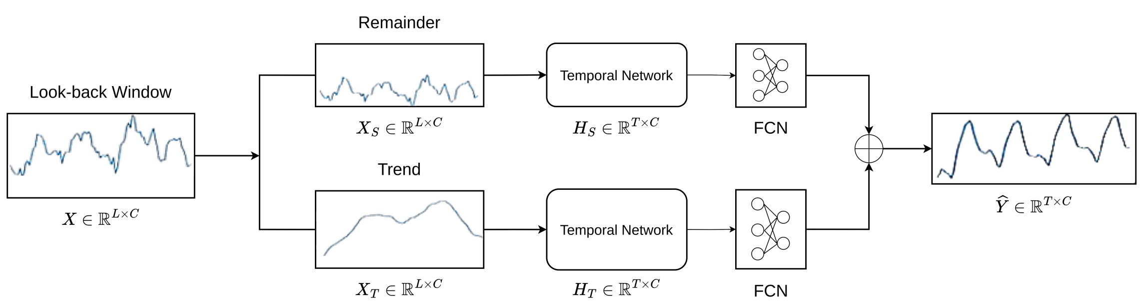


Figure 5: Complete Model Structure

4 Testing and Experiments

This project employed Keras with TensorFlow to create the deep neural network models. The experiment was conducted multiple times on data from four cities to fine-tune the hyper-parameters. We configured hyperparameter settings for our models to achieve the optimal predictive performance as shown in table 1.

In order to quantitatively analyze the accuracy, mean squared error (MSE) and mean absolute error (MAE) are used to evaluate the performance of the model.

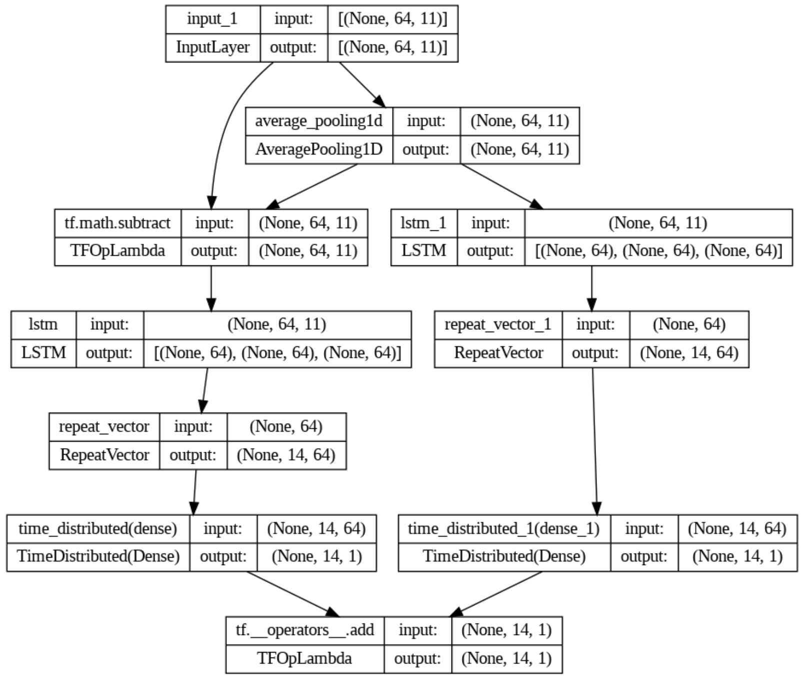


Figure 6: LSTM based Temporal Network

Table 1: Hyperparameters

Hyperparameter	Value
Optimizer	Adam
Loss	Mean Squared Error
Batch Size	32
Epochs	10
Window Size	64
Scaler	MinMax

5 Results and discussion

The errors we get after training on the test set for both models are shown in the tables 2 and 3. The results shown are for step size = 14.

6 Conclusion

In this report, $PM_{2.5}$ concentration forecasting models are developed with the LSTM network and LSTM auto-encoder network. For multi-step ahead prediction, the proposed model achieved satisfactory results for both short and long-term prediction tasks.

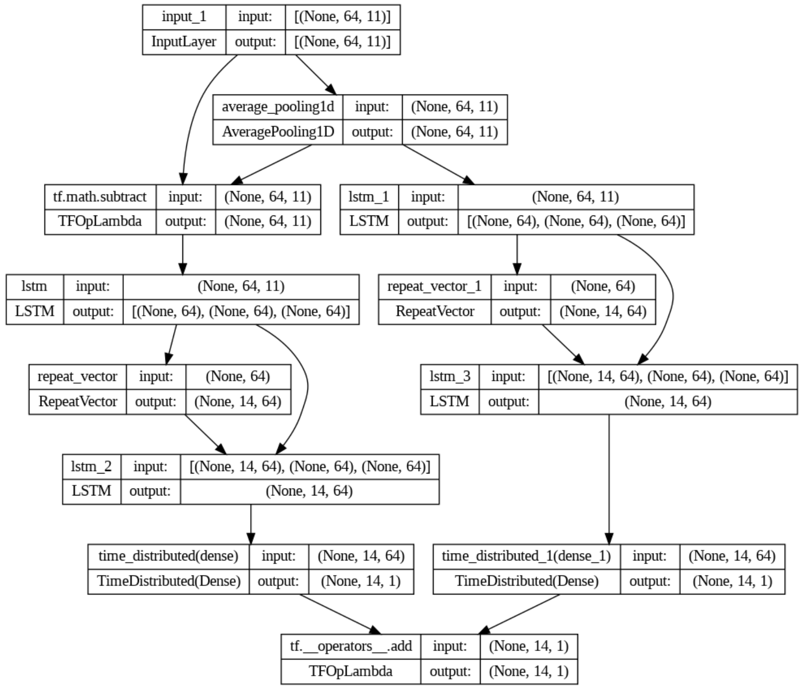


Figure 7: LSTM auto-encoder based Temporal Network

Table 2: Results for LSTM-based model

City	Mean Squared Error	Mean Absolute Error
B	0.0113	0.0768
G	0.0069	0.0577
S	0.0063	0.0604
T	0.0057	0.0563

Table 3: Results for LSTM auto-encoder based model

City	Mean Squared Error	Mean Absolute Error
B	0.0119	0.0749
G	0.0071	0.0578
S	0.0059	0.0590
T	0.0060	0.0575

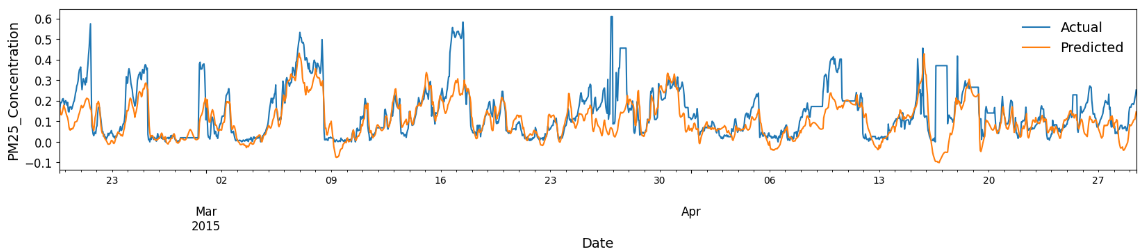


Figure 8: Actual vs Predicted Graph for City B

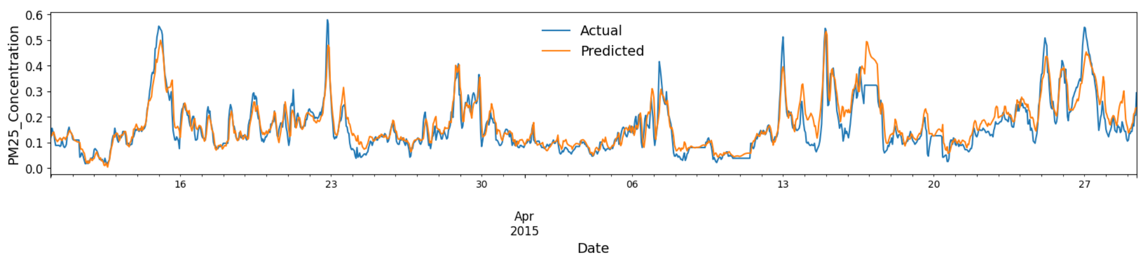


Figure 9: Actual vs Predicted Graph for City G

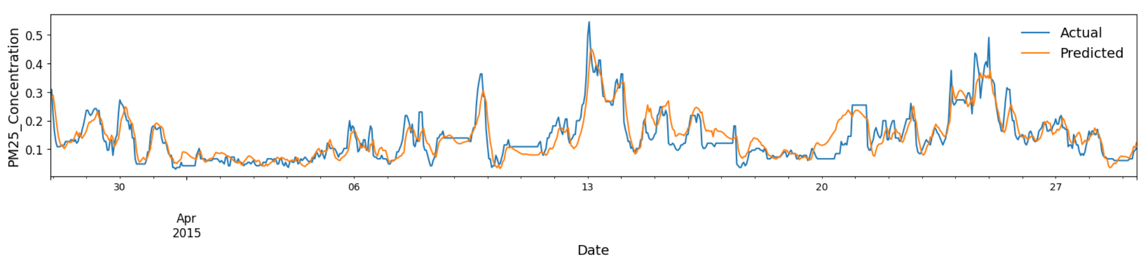


Figure 10: Actual vs Predicted Graph for City S

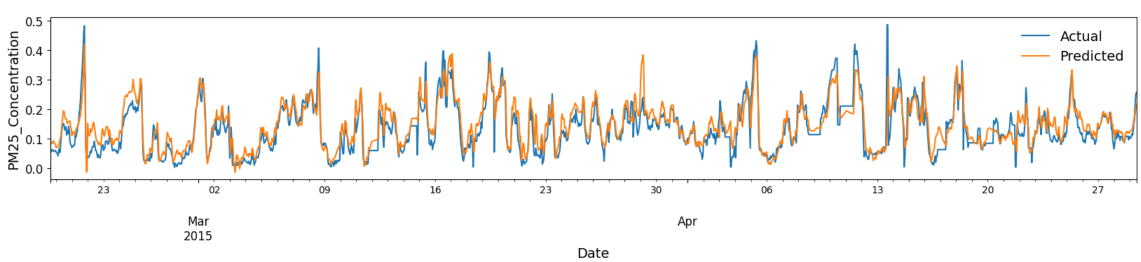


Figure 11: Actual vs Predicted Graph for City T