METHODOLGY

`

2 Chapter 2

2.1 Methodology

2.1.1 Site and Data Description

In order to facilitate broad agricultural planning and the creation of long-term strategies, India has been divided into fifteen major agro-climatic zones (Figure 2-1) based on climate, geological formation, physiography, cropping patterns, etc. Therefore, in order to ensure that the data accurately represented the Indian subcontinent, every agroclimatic zone in India was tried to cover for our research work. As of the time of our research, there were no data available for two regions out of the fifteen agroclimatic zones: the Island Region of India (Andaman & Nicobar Island, Lakshadweep Island) and the Western Himalayan Division (Union Territories of Jammu and Kashmir and Ladakh). As a result, they were excluded from the present study. Depending on the availability of data, the data was collected from 1st January 2015 to 31st May 2020.

`

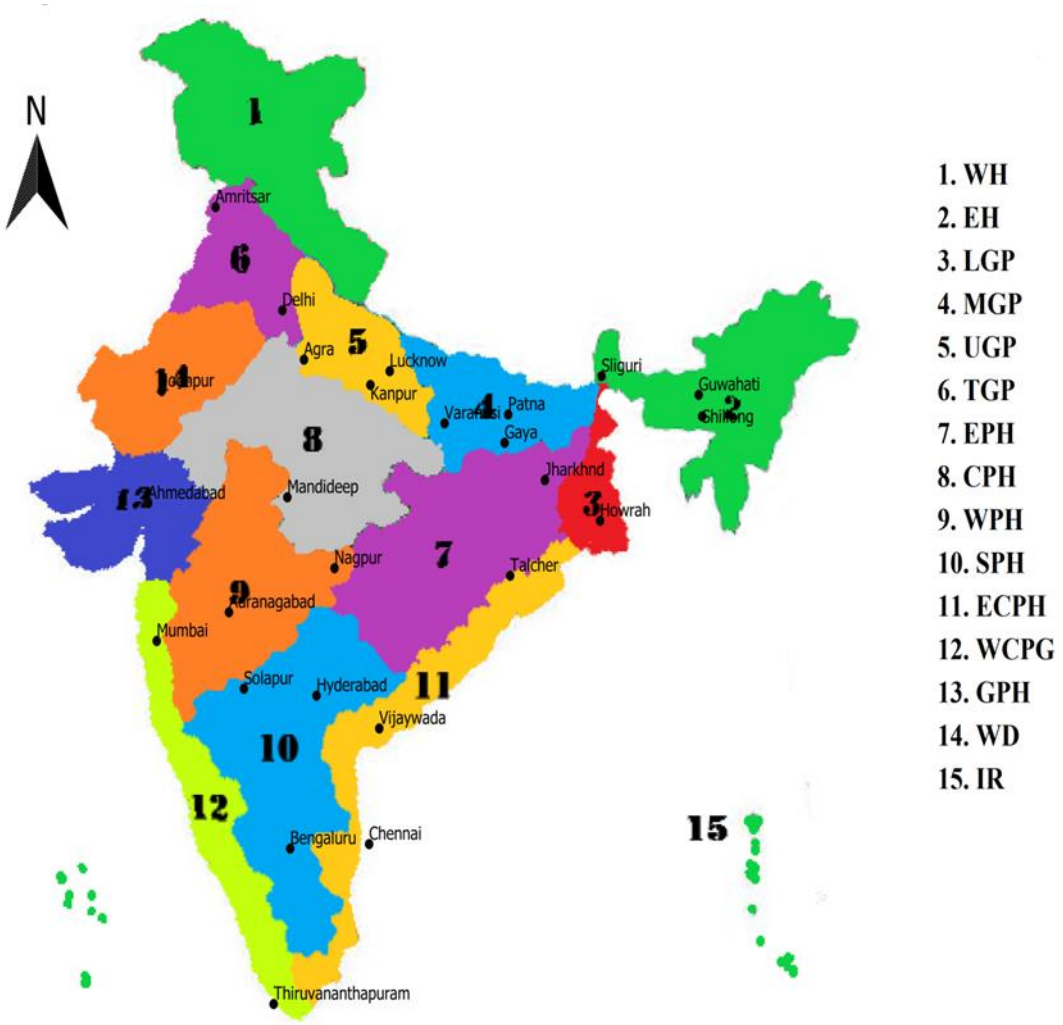

Figure 2-1 Agro climatic zones of India [Image source: https://vikaspedia.in/agriculture/crop-production/weather-information/agro-climatic-zones-in-india]

Central Pollution Control Board (CPCB) of India, a Government agency has provided us with data from all over India for this purpose (http://www.cpcb.nic.in/). Data from State level Pollution Control Boards of India are also provided by the CPCB. From all over India, 26 monitoring stations were selected for data retrieval. Details of the data used are described in Table 2-1.

`

Table 2-1: Data Description

Sl. No.	Agroclimatic Zone	Zone Code	City and Station Name	Data Source	Data Period	
					From	To
1.	Eastern Himalayan Region	EH	**Shillong** (Lumpyngngad)	Meghalaya PCB	27August 2019	31 May 2020
			Guwahati (Railway Colony)	Assam PCB	16 February 2019	31 May 2020
			Sliguri (Ward 32, Bapupara)	WBPCB	1 February 2018	31 May 2020
2.	Lower Gangetic Plain Region	LGP	**Howrah** (Padmapukur)	WBPCB	19 January 2018	31 May 2020
3.	Middle Gangetic Plain Region	MGP	**Gaya** (Collectorate)	BSPCB	1 January 2016	31 May 2020
			Patna(IGSC Planetarium Complex,)	BSPCB	21 October 2017	10 April 2020
			Varanasi (Ardhali Bazar)	UPPCB	1 January 2015	31 May 2020
4.	Upper Gangetic Plains Region	UGP	**Agra** (Sanjay Palace)	UPPCB	11 May 2015	31 May 2020
			Kanpur (Nehru Nagar)	UPPCB	12 May 2015	31 May 2020
			Lucknow (Central School)	CPCB	28 March 2015	31 May 2020
5.	Trans-Ganga Plains Region	TGP	**Amritsar** (Golden Temple)	PPCB	27February 2017	31 May 2020
			Delhi (ITO)	CPCB	3 November 2016	31 May 2020
6.	Eastern Plateau and Hills	EPH	**Jamshedpur** (Tata Stadium, Jorapokhar)	JSPCB	9 January 2019	31 May 2020
			Talcher (Talcher Coal Fields)	OSPCB	7 February 2018	31 May 2020
7.	Central Plateau and	CPH	**Mandideep** (Sector D)	MPPCB	2 January 2018	31 May 2020

`

	Hills		**Nagpur** (Opposite GPO Civil lines)	MPCB	30 March 2016	31 May 2020
8.	Western Plateau and Hills	WPH	**Aurangabad** (More Chowk Waley)	MPCB	1 October 2017	31 May 2020
			Mumbai (Bandra)	MPCB	6 May 2018	31 May 2020
			Solapur (Solapur)	MPCB	1 January 2016	31 May 2020
9.	Southern Plateau and Hills	SPH	**Bengaluru** (Peenya)	CPCB	23 March 2015	31 May 2020
			Hyderabad (Zoo Park)	TSPCB	1 October 2015	31 May 2020
10.	Eastern Coastal Plains and Hills	ECPH	**Chennai** (Manali)	CPCB	23 March 2015	31 May 2020
			Vijayawada (PWD Grounds)	APPCB	26 April 2017	27 October 2019
11.	Western Coastal Plains and Ghats	WCPG	**Thiruvananthapuram** (Plammoodu)	KPCB	21 June 2017	31 May 2020
12.	Gujarat Plains and Hills	GPH	**Ahmedabad** (Maninagar)	GPCB	1 January 2018	31 May 2020
13.	Western Dry Region	WD	**Jodhpur** (Collecorate)	RSPCB	21 September 2017	31 May 2020
14.	Western Himalayan Division	WH	No data available during the period			
15.	Island Region	IR	No data available during the period			

Wind Characteristics of the region:

In an area close to a source of air pollution such as industries, heavy vehicular transport, mining etc., air movement plays a major role in the increase of air pollution in that area. Local air quality typically changes over time due to the influence of weather patterns. In stable conditions, or when there is little to no vertical air

`

movement, air pollutants can build up close to the ground and result in severe outbreaks of air pollution. The Windrose diagram gives us a picture of the wind pattern in a particular area. Windrose diagrams were plotted for each agroclimatic zones using the average data (Figure 2.2 to Figure 2.14).

Eastern Himalayan Region (EH):

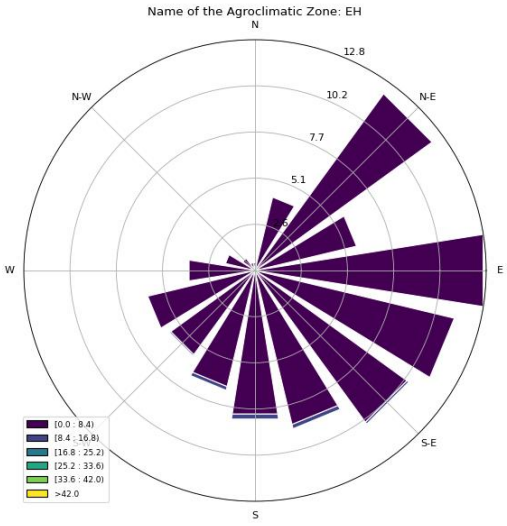

Figure 2-2: Windrose Plot of EH

Lower Gangetic Plain Region (LGP):

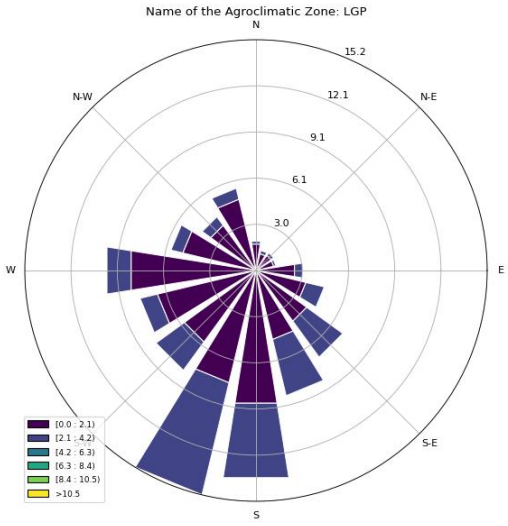

Figure 2-3 Windrose Plot of LGP

Middle Gangetic Plain Region (MGP):

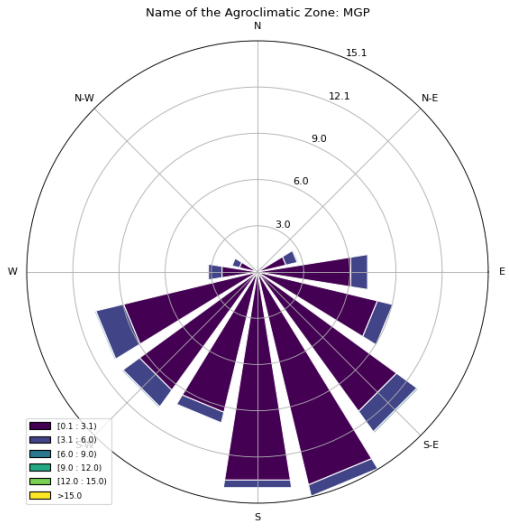

Figure 2-4 Windrose Plot of MGP

Upper Gangetic Plains Region (UGP):

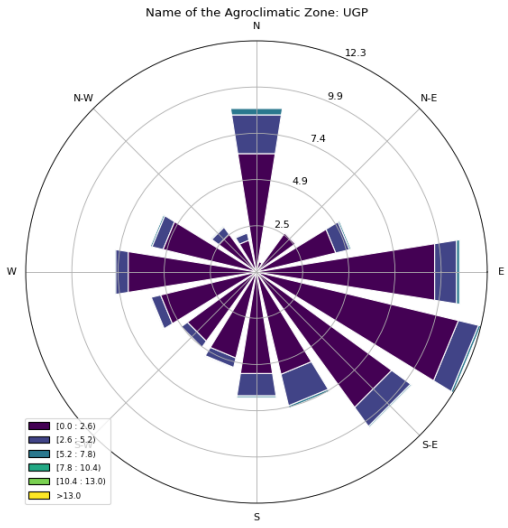

Figure 2-5 Windrose Plot of UGP

Trans-Ganga Plains Region (TGP):

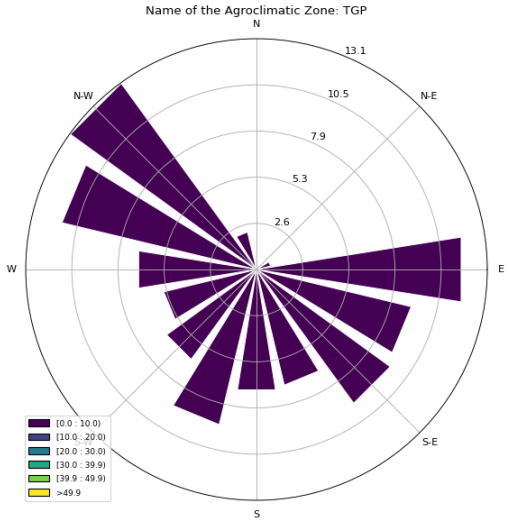

Figure 2-6 Windrose Plot of TGP

Eastern Plateau and Hills (EPH):

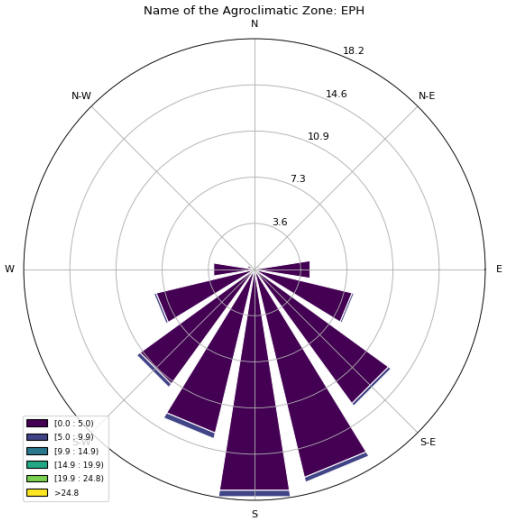

Figure 2-7 Windrose Plot of EPH

Central Plateau and Hills (CPH):

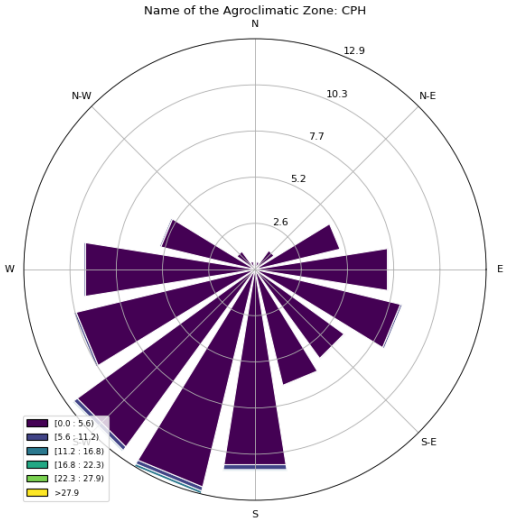

Figure 2-8 Windrose Plot of CPH

Western Plateau and Hills (WPH):

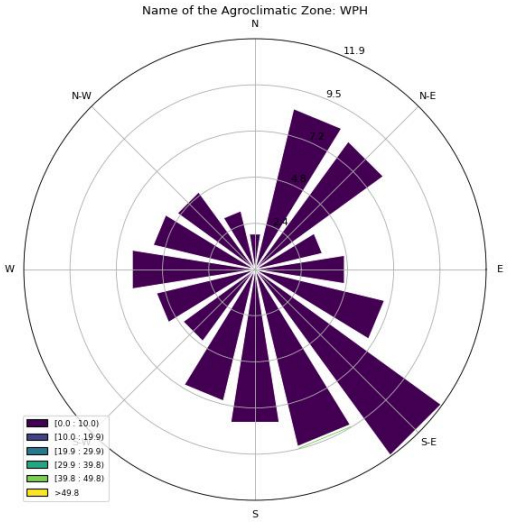

Figure 2-9 Windrose Plot of WPH

Southern Plateau and Hills (SPH):

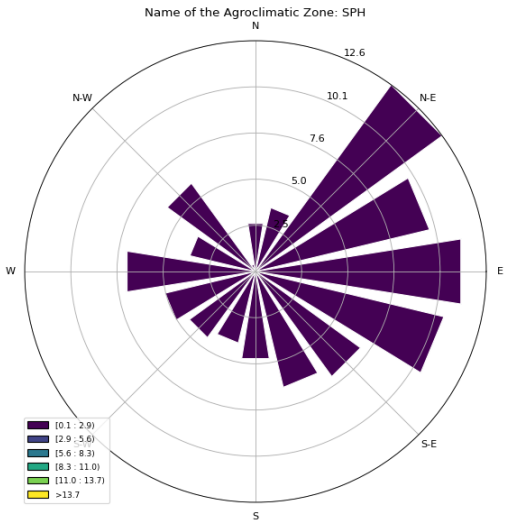

Figure 2-10 Windrose Plot of SPH

Eastern Coastal Plains and Hills (ECPH):

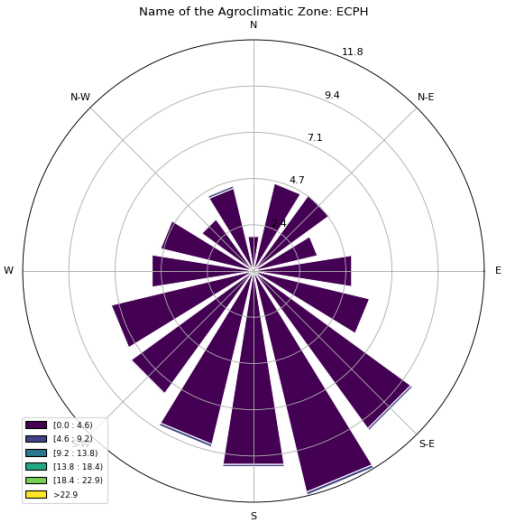

Figure 2-11 Windrose Plot of ECPH

Western Coastal Plains and Ghats (WCPG):

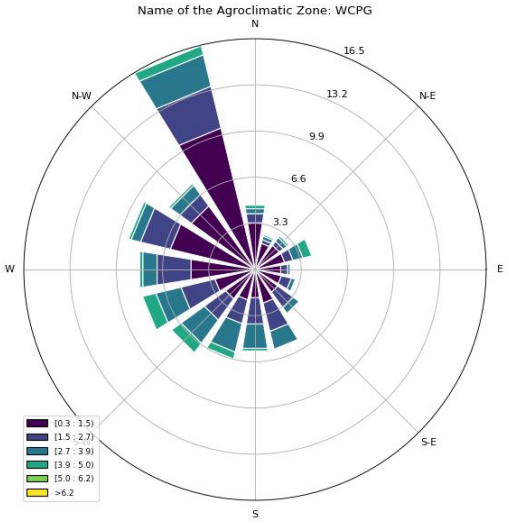

Figure 2-12 Windrose Plot of WCPG

Gujarat Plains and Hills (GPH):

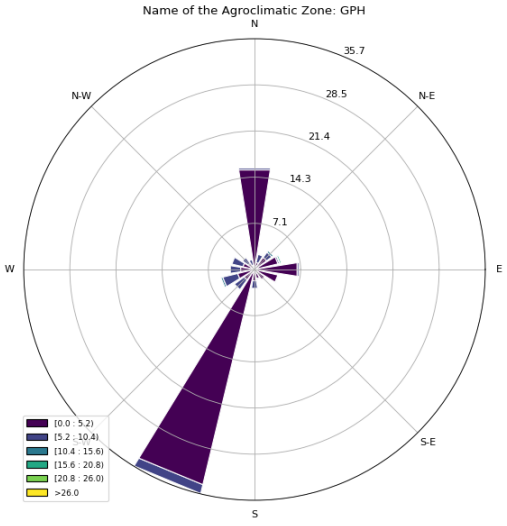

Figure 2-13 Windrose Plot of GPH

Western Dry Region (WD):

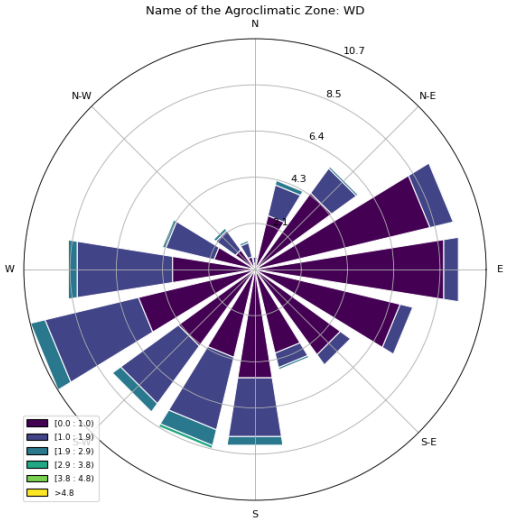

Figure 2-14 Windrose Plot of WD

2.1.2 Data Pre-processing

Missing values and outliers were found to be common in all secondary data sources. Therefore, data pre-processing is essential to minimize and eliminate these errors. In the present study, values that were excessively high were classified as outliers. Linear interpolation method [114] was used to replace the outliers. Missing values present in the dataset were also filled with same technique.

Figure below demonstrates the box plot of pre-processed data acquired from the monitoring stations:

`

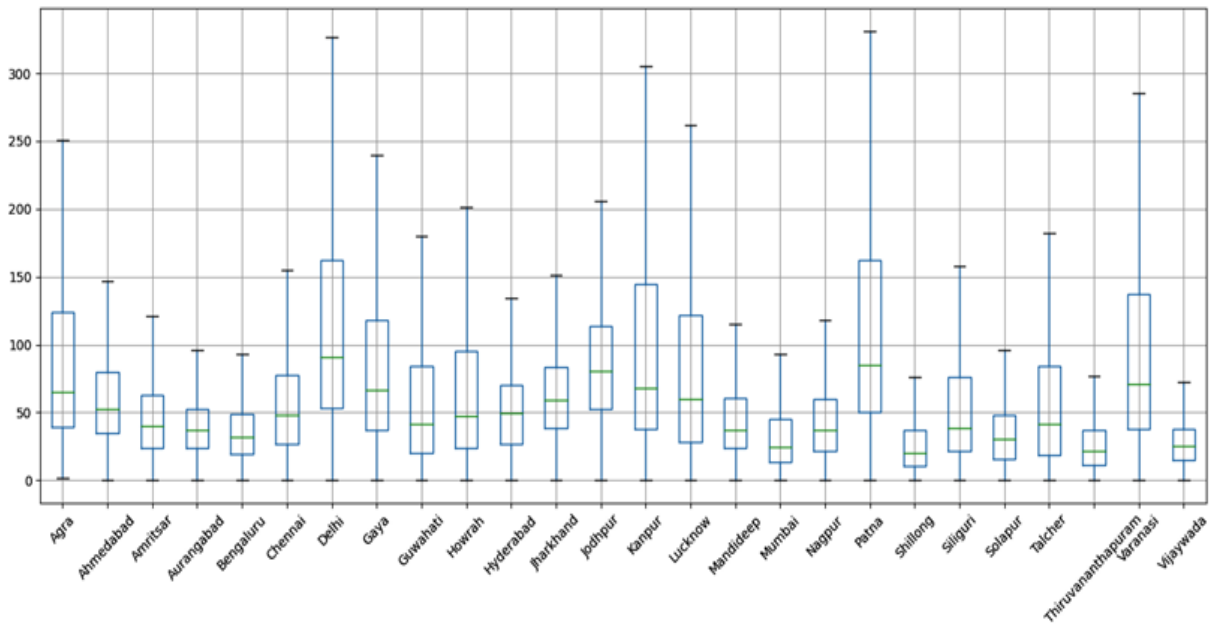

Figure 2-15 Box plot of Data

The average value of 26 monitoring stations:

Count	Mean	Standard Deviation	Min	25%	50%	75%	Max
26	65.12	28.97	25.49	42.20	58.70	89.05	119.53

The detail average data distribution is described in Figure 2-16

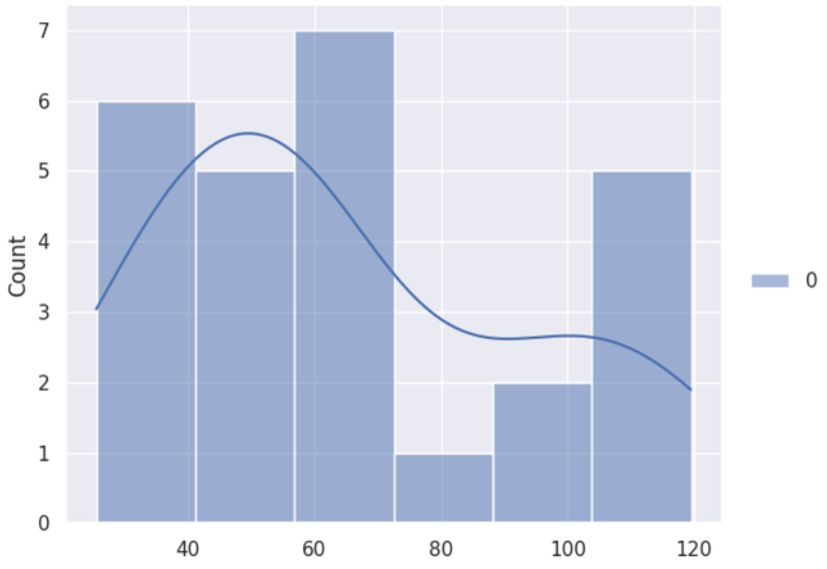

Figure 2-16 Average distribution of Data

`

North Indian cities have relatively high value of $PM_{2.5}$ concentration as compared to rest of the India data. On the other hand Eastern cities have low concentration of $PM_{2.5}$value. Line plot of processed data for a typical monitoring station is depicted in Figure 2-17.

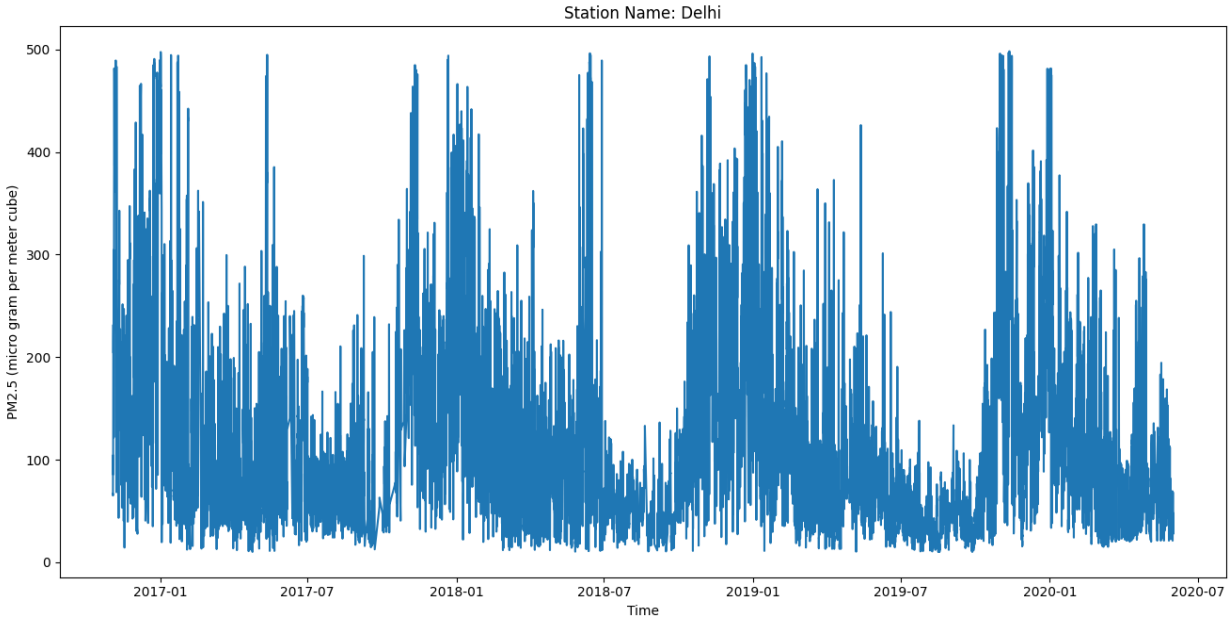

Figure 2-17 Line plot of Delhi data

2.1.3 Sequence to sequence modeling

In machine learning, predicting the next symbol or symbols based on the previously observed sequence of symbols is called sequence prediction or sequence to sequence modeling (Seq2Seq). These symbols could be an object, product, an event, an alphabet, a word, or a number. In contrast to other supervised learning problems, sequence prediction requires that the data order be maintained during model training and prediction. Here both the input and output sequences can have different lengths. Seq2Seq models are trained using a dataset of pairs just like any other supervised learning model. Concept of sequence to sequence modeling and encoder-decoder architecture was used in this paper (Figure 2-18). Sequence prediction was a challenging task and its presence could be found for a long time. [115] for the first time mapped the whole input sentence into a vector. [116] proposed seq2seq model

`

for machine translation. [117] used Seq2Seq model for attention-based air quality predictor. An encoder decoder model architecture consists of two parts and one intermediate phase to construct a context vector. After the input sequence has been read and encoded, the decoder decodes it and predicts each element of the output sequence. In our model a series of ConvLSTM followed by a 3D CNN acts as an encoder and extracts features from the input sequences. A series of BLSTM acts as a decoder and generates predicted output sequences. BLSTM model used in the decoder, keeps track of prior hour prediction in the sequence and stores the information as an internal state for generating output sequence.

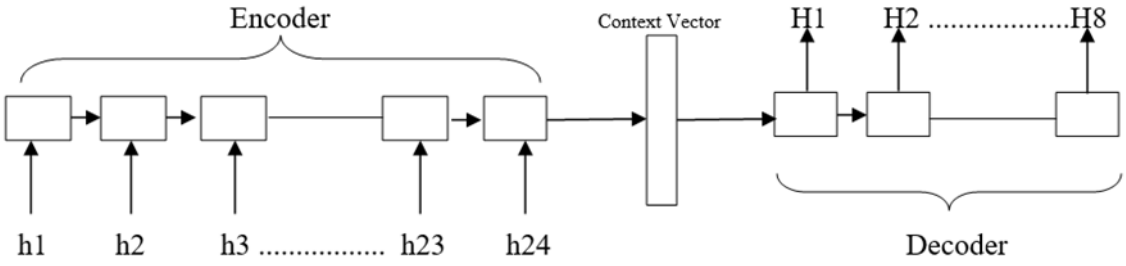

Figure 2-18 Seq2Seq model

2.1.4 Recurrent Neural Network (RNN)

RNN is a type of artificial neural network with inherent memory that uses past information to predict the next step [118].Three layers make up a basic RNN: input, hidden, and output layers. The hidden states provide a prediction at the output layer for each timestep based on the input vector. For an RNN, the hidden state is a set of values that collectively, regardless of outside factors, contain all the unique data needed to reconstruct the network's prior states over multiple time-steps. Accurate predictions at the output layer can be made using this integrated information to define the network's future behavior. The hidden state of a typical RNN (Figure 2-19) at time t can be explained with the following equation:

$$h_t = tanh(W * [h_{t-1}, x_t] + b)$$

`

where h_t and h_{t-1} are hidden states representing current and previous time steps, respectively. x_t is the input vector, W is the weight and b is the bias that are shared among different time steps.

The following equation yields the final network output:

$$f = W_f * h_{t-1} + b_f$$

Where f is the final state.

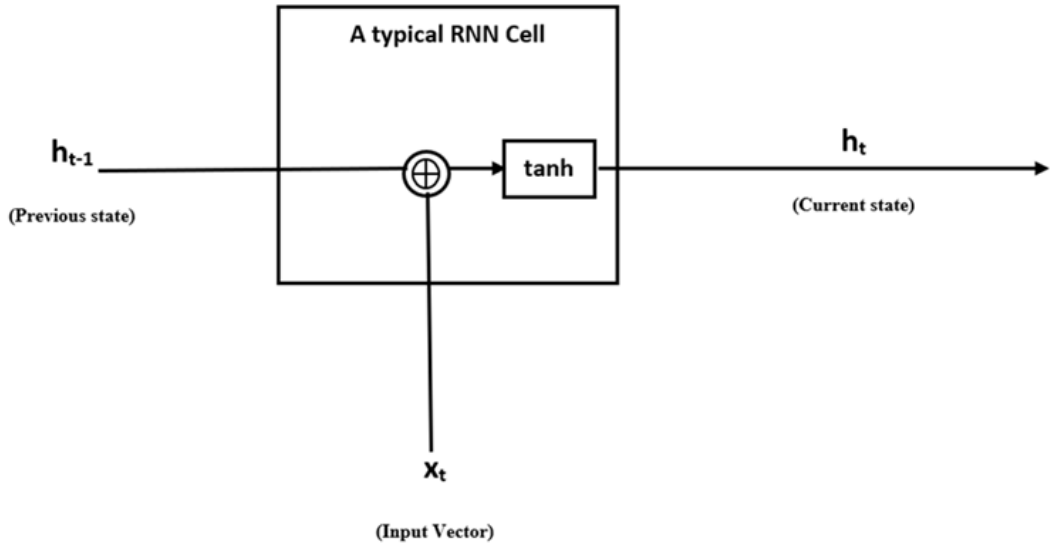

Figure 2-19 RNN

2.1.5 Long Short-Term Memory (LSTM)

The LSTM is a type of recurrent neural network that is specifically designed to handle long-term dependencies within time series [119,120]. RNN could store and retrieve the past information from their internal cycles. However, LSTM can control the flow of information across its internal states and cells by allowing an information to pass, store or delete at any moment of time with the help of some self-controlling gates. Figure 2-20 demonstrates the basic architecture of a LSTM cell that consists of four elements- (i)Input gate (i_t), (ii)Output gate (o_t), (iii) Forget gate (f_t) and (iv)Cell status ($Cell_t$). New information could be stored in the LSTM cell by activating the input gate. Stored information could be deleted from memory by activating the forget gate. One could control the flow of current cell information by activating or deactivating

`

the output gate. Error or gradients produced in the LSTM cell were trapped inside the cell with the help of controlling gates. Thus, error neither vanishes nor explodes rapidly as observed in traditional artificial neural networks.

Equations in LSTM operations are:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma\big(W_f * [h_{t-1}, x_t] + b_f\big)$$
$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$
$$\tilde{C_t} = tanh(W_c * [h_{t-1}, x_t] + b_c)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C_t}$$
$$h_t = o_t * tanh(C_t)$$

where i_t=Input gate, f_t=Forget gate, o_t=Output gate, C_t=Current Cell state, $\tilde{C_t}$ =Updated Cell State, C_{t-1} is the previous cell value, h_t= Current state, h_{t-1} the previous hidden state. tanh () is the activation function and σ the logistic sigmoid function. $b_{i\ (/f/o/c)}$ represents corresponding bias vector; $W_{i\ (/f/o/c)}$ represents the input weight.

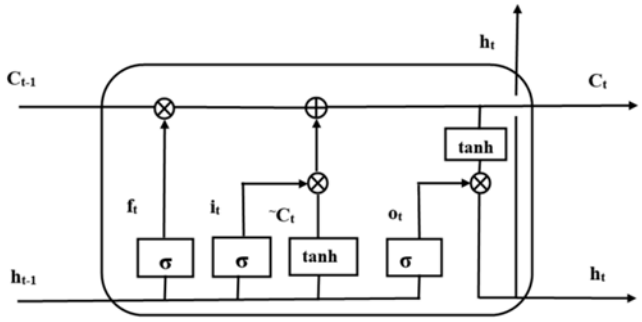

Figure 2-20 LSTM Network

2.1.6 Bidirectional-LSTM (BLSTM)

BLSTM is an enhanced version of LSTM that can process the information from both directions. Essentially, it can memorize the present and past information simultaneously in its structure [121]. Input sequences would be trained by two LSTMs from beginning to end (forward direction) and end to beginning (backward direction), thus feeding the additional context into the network. At any moment of time BLSTM can preserve past and future values by combining two sets of LSTMs. This increases the memory and the performance of the network. The outcome of BLSTM network can be represented by the following equation:

`

$$h_{bi(t)} = \overrightarrow{f_{(t)}} + \overleftarrow{b_{(t)}}$$

Where $f_{(t)}$ and $b_{(t)}$ are results obtained from forward and backward LSTM respectively.

2.1.7 Three Dimensional Convolutional Neural Network (3D CNN)

A typical CNN model is a stack of many convolutional layers with 2D convolutional kernels. They were designed for analysis of 2D objects. Most of the real-world situations were 3D in nature and traditional CNN could not capture the third dimension. To overcome this issue 3D convolution was proposed by [122] for video classification which can extract features in space and time dimension. Partial connectedness and weight sharing were the two basic properties of CNN (Figure 2-21). Partial connectedness decreases the chance of overfitting in CNN and preserve local spatial features. Hence CNN are suitable for short range predictions. The weights sharing reduces the number of parameters and enhance network generalization capability. In this paper we have used 3DCNN for capturing short range features of $PM_{2.5}$concentration present in the series. By adjusting the number of 3D kernels and feature maps present in each layer we can extract deep features and avoid overall complexity of the model [123].

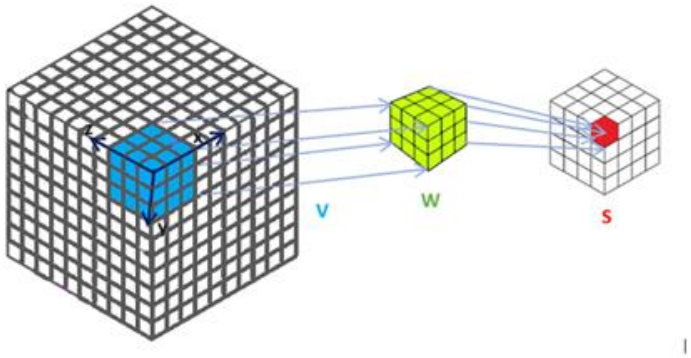

Figure 2-21 3D CNN

2.1.8 Convolutional LSTM (ConvLSTM)

ConvLSTM is a special kind of LSTM where convolutional operators are used in place of fully connected layer operators [124]. Here all inputs, cell outputs, hidden states and gates were considered as 3D tensors and resembled like vectors inside a 3D

`

grid structure. Last two dimensions of the 3D tensors were considered as spatial (or second) dimensions. In the grid, next state of a cell is determined with input values and last states of local surroundings with the help of convolution operation in state to state and input to state transitions. Equations involved in ConvLSTM operation are like LSTM operations but with a convolutional operator. Figure 2-22 represents the internal structure of a ConvLSTM. An input value X_t was used to calculate present state h_t at time t, with past state h_{t-1} and next state h_{t+1}.In a similar fashion the next present state was calculated.

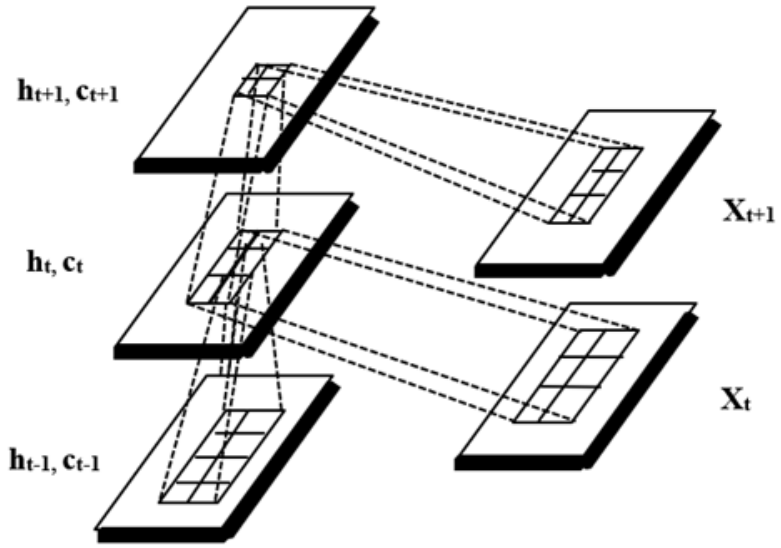

Figure 2-22 ConvLSTM

2.1.9 Bi-Convolutional LSTM (BConvLSTM)

The BConvLSTM is an extended version of ConvLSTM where two states of a sequence are maintained at a time: one in forward direction and other in backward direction. In each LSTM cell two numbers of cell and hidden states are monitored. Thereby, BConvLSTM can access more information and yields better performance than ConvLSTM [125].

Figure 2-23 illustrates the working principle of a BConvLSTM cell. It consists of a ConvLSTM cell having two sets of hidden state and cell state. One set (h_f , c_f) is used

`

for forward direction and the other set (h_b, c_b) is used for backward direction. Corresponding hidden states from each state in a given time step, are stacked and transferred through the convolution layer to obtain final hidden representation of that time step. Next layer of the BConvLSTM receives this final hidden representation as an input.

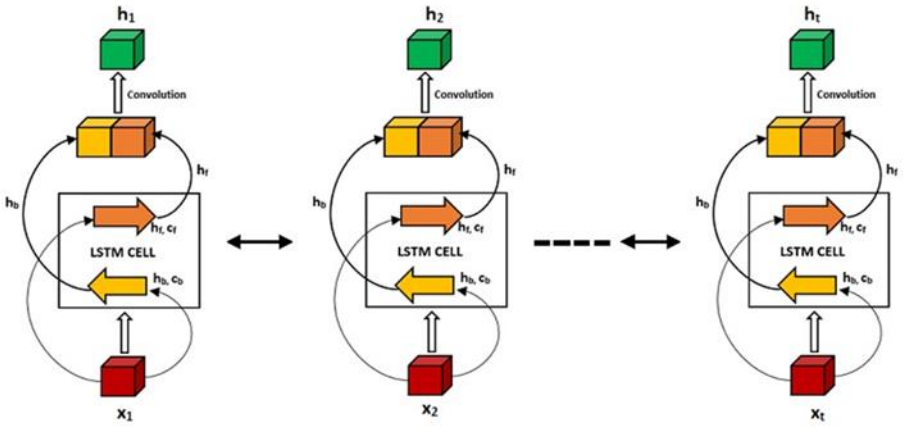

Figure 2-23 BConvLSTM

`