Presentation

on

Forecasting of Time Series Data in Python

A Component for the Fulfillment of the Course - Practice School I of
Birla Institute Of Science and Technology, Pilani

Presented by-

01 Swadesh Vaibhav

2017A7PS0030P

02 Gandhi Atith

2017A7PS0062P

Mentors-

01 **Dr. Mayank Goel**

Practice School Mentor

02 **Mr. Sakthivel Samy V**

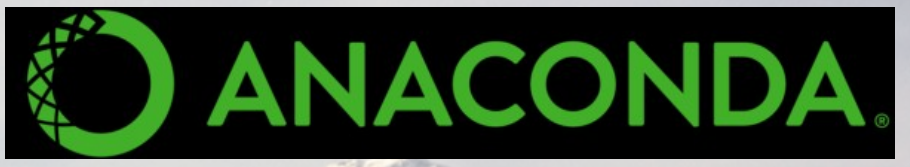
Scientific Mentor

About NCPOR

- National Centre for Polar and Ocean Research (NCPOR), erstwhile National Centre of Antarctic and Ocean Research, is India’s premier R&D institution responsible for the country’s research activities in the polar and Southern Ocean realms.

- NCPOR was established as an autonomous Research and Development Institution of the Ministry of Earth Sciences (formerly Department of Ocean Development), Government of India on the 25th May 1998.

Softwares and Libraries Used



Objectives:

Descriptive Analysis

Analysing the given data to draw inferences.

Predictive Analysis

Forecasting the future values of these datasets by using various Statistical and Machine Learning Predictive Techniques.

Datasets Available

BHARATI STATION
IIG
IMD

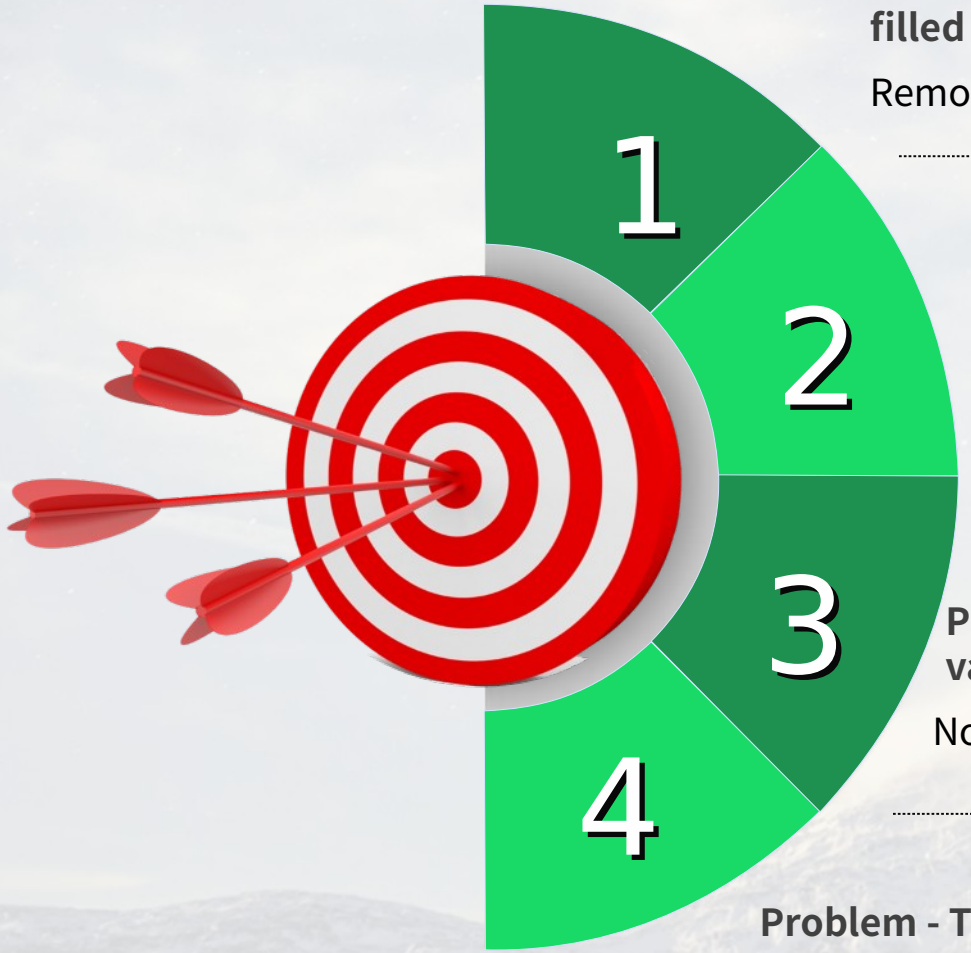
MAITRI STATION
IIG
IMD
Sankalp
Dozer

Parameters

The following parameters were to be analysed in the given datasets:

Temperature : Range - Generally sub-zero
Air Pressure : Range - Generally between 700 and 900
Relative Humidity : Range - Between 0 and 100
Wind Speed : Range - Above 0

Data Preprocessing



Problem - The Data is missing at places, whose entries have been filled by the standard value, -999.

Remove these data points before applying the prediction models.

Problem - The data has anomalous points which will reduce the efficieny of the training algortihms.

Remove these points through anomaly detection

Problem - The parameters show variation in terms of the range of values, which would decrease the performance of the model.

Normalise the data before training.

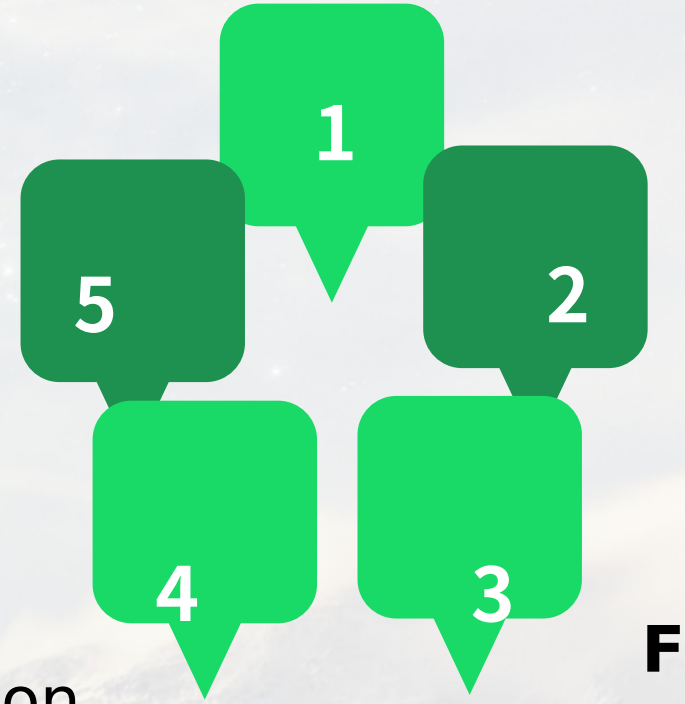
Problem - The learning problem is unsupervised.

Create a moving window of 12 time steps (for monthly data) while training.

Prediction Models:

LSTM - Long Short Term Memory

RNN - Recurrent Neural Network



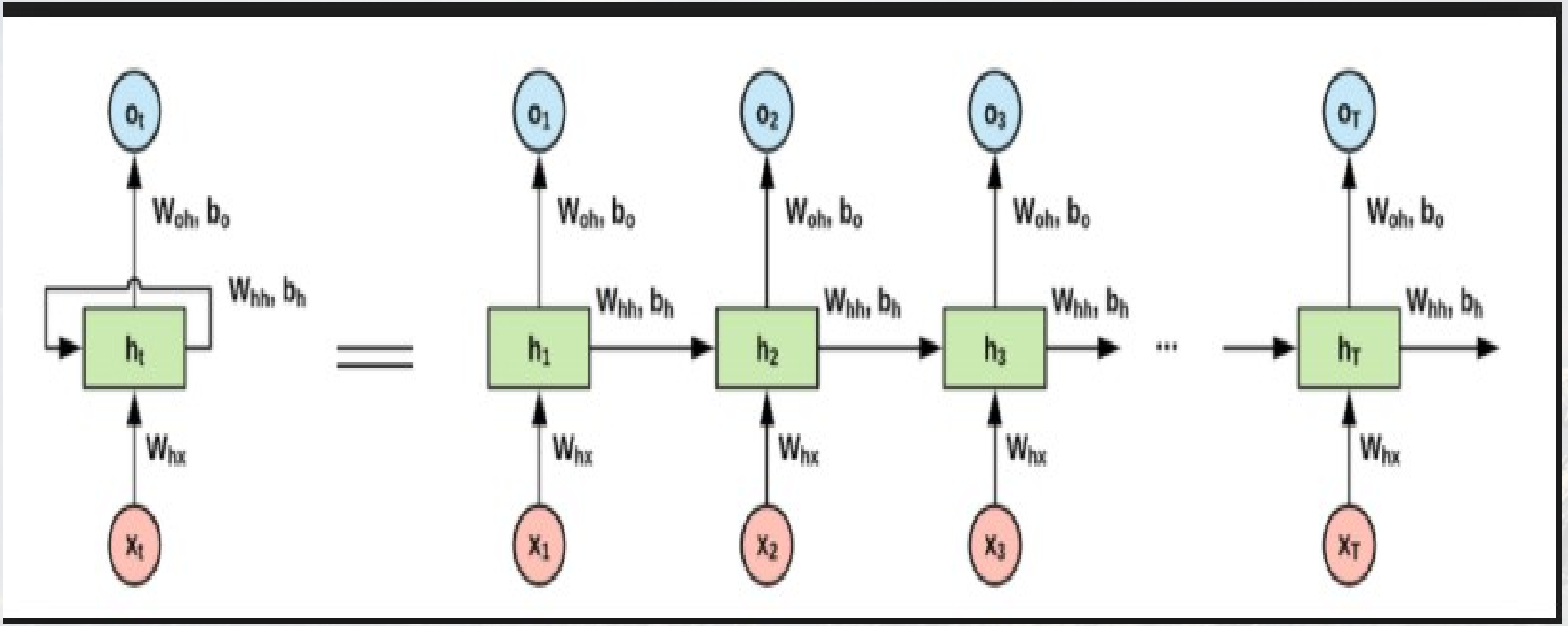
CNN - Convolutional Neural Network

ARIMA - AutoRegression, Integration, Moving Averages

FBProphet

What is RNN?

Recurrent Neural Network



RNN

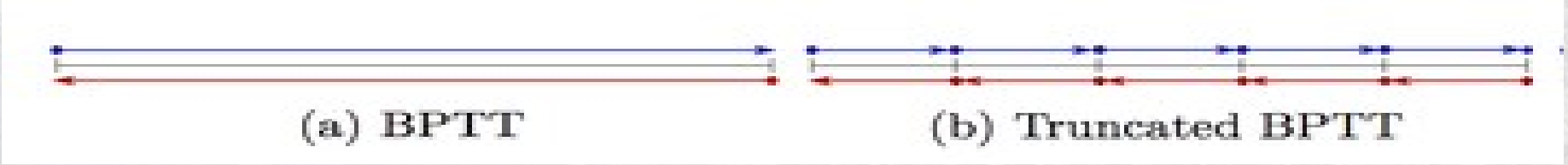
- RNN differs from other feedforward neural network in the sense that it co nsiders the previous state before calculating the output of the next state.
- It takes a input and its previous state , updates its state and gives output .
- As time series data depends upon its previous state it is widely used in ti me series analysis.
- It basic equation is :

  h(t)=f(h(t-1),x(t)) where h(t) is the new state at time t and x is input at tim e t.

  y(t)=W*h(t) where y(t) is the output.

Training of RNN

- It trains through a method known as Backpropagation Through Time.
- BPTT works by unrolling all input timesteps. Each timestep has one input timestep, one copy of the network, and one output. Errors are then calcul ated and accumulated for each timestep. The network is rolled back up an d the weights are updated.
- If the data is very large , Truncated BPTT is used.
- TBPTT, is a modified version of the BPTT training algorithm for recurrent n eural networks where the sequence is processed one timestep at a time a nd periodically (k1 timesteps) the BPTT update is performed back for a fix ed number of timesteps (k2 timesteps).



Problems with RNN

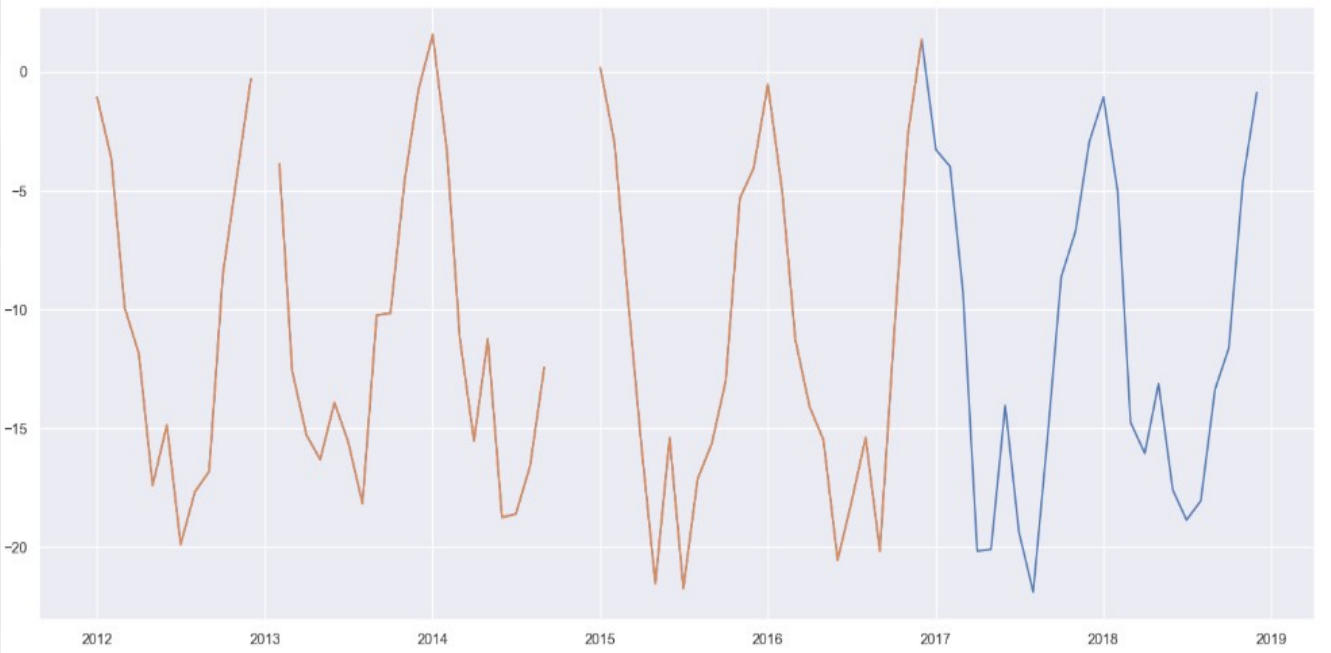
- As RNN unrolls during Backpropagation it requires a lot of memory and C omputational power, as it is equivalent to do backpropagation on t feedfo rward networks.(t is the number of timesteps).
- Thus it also suffers from vanishing gradient problem exponentially.
- So, for large data RNN is not very good and so models with Gated memor y like GRU and LSTM are used for large datasets.

RNN model Used

- A SimpleRNN model of keras was used.
- A layer of 50 nodes was used in the model

```
model = Sequential()
model.add(SimpleRNN(50, input_shape=(n_steps, n_features), return_sequences=False,activation='tanh'))
model.set_weights([np.random.rand(*w.shape)*0.2 - 0.1 for w in model.get_weights()])
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

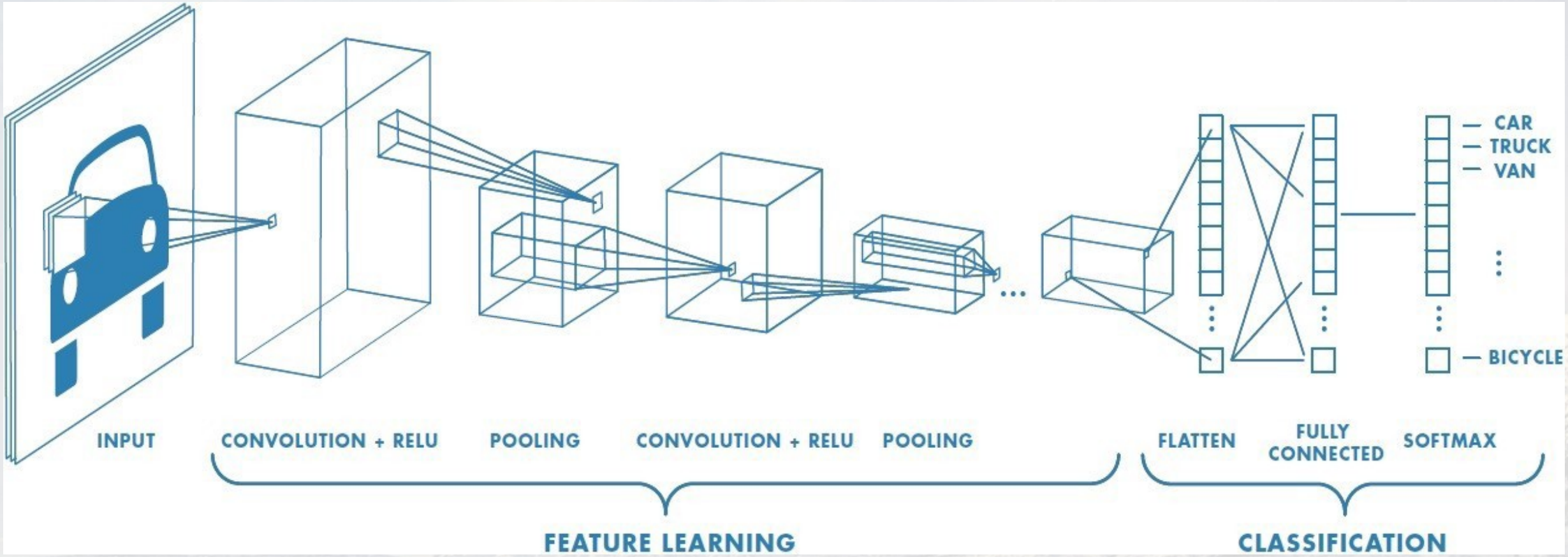
Result

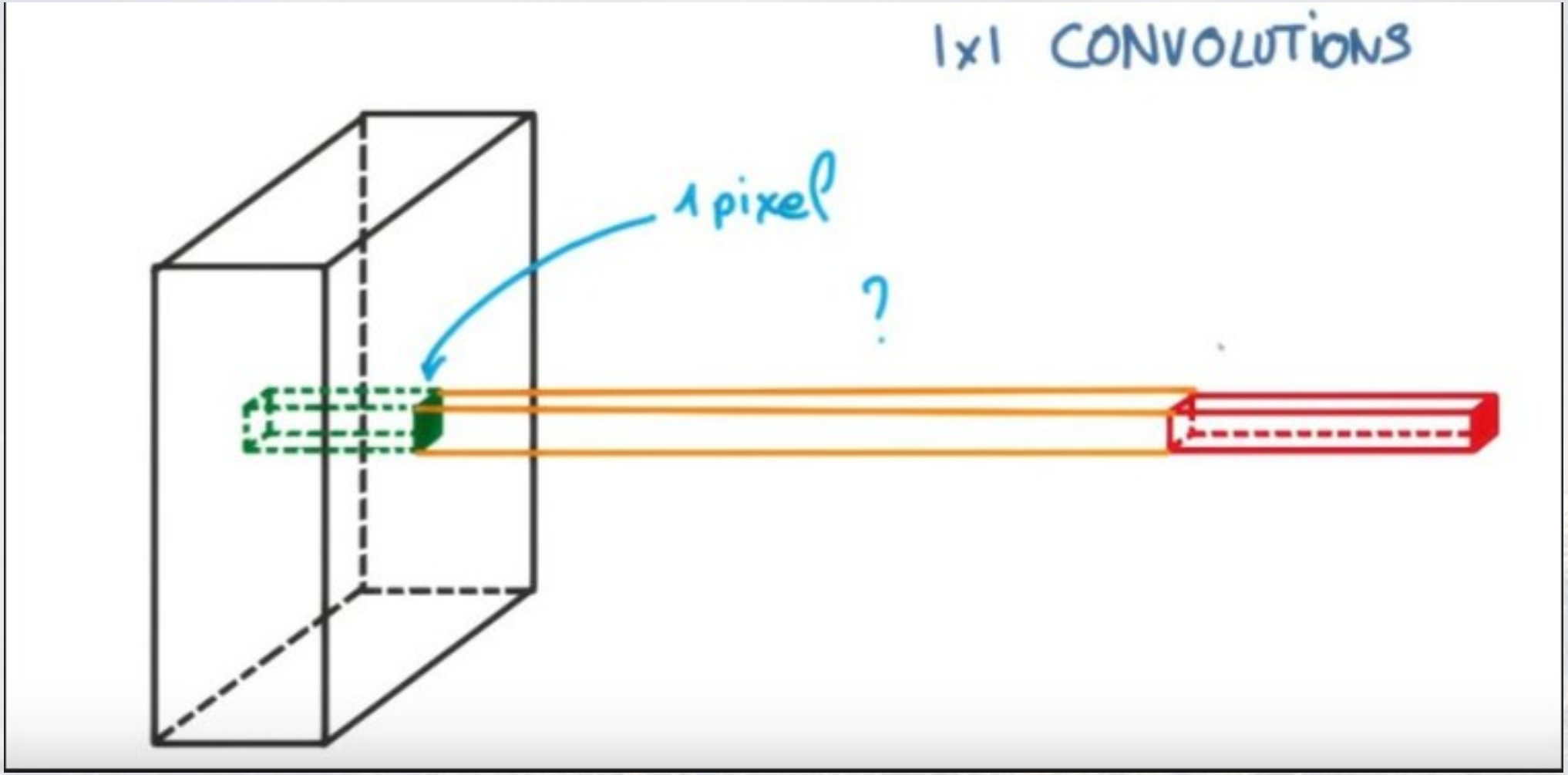


Model - RNN
Dataset - IIG Bharati
Parameter - Temperature
Duration - 2 years

What is CNN?

Convolution Neural Network




1x1 CONVOLUTIONS
1 pixel
?

CNN for time series analysis

- Ideally CNN is used for Image processing as it is good for recognising patte rns but we implemented CNN for finding patterns in time series data of th e various datasets.
- It generally has 3 layers:

  1.Convolution layer which using various filters recognises the patterns in t he data.

  2.Max Pooling layers which reduces the size of the data without affecting the patterns in the data.

  3.Fully connected layer which works like a regular feedforward neural net work and is connected to output layer.

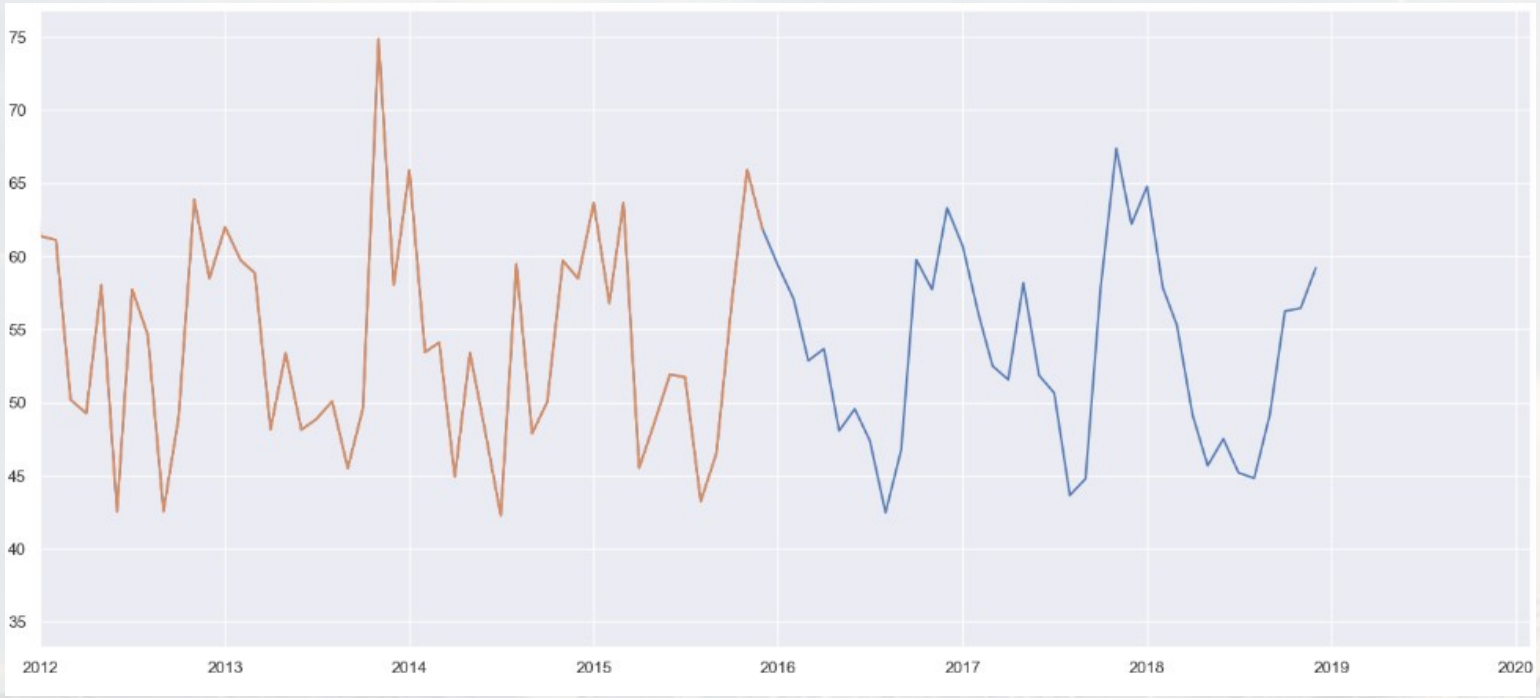
CNN model used

- As the data was time series and not image Conv1D model was us ed.
- The numbers of filters used were 64 and the kernel size of the fil ter was 2.
- A Maxpooling layer with stride or pool-size 2 was used and finall y a Flatten layer was used which was connected with output laye r.

CNN model

```
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(n_steps, n_features)))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(50, activation='relu'))
model.add(Dropout(0))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

Result



Model - CNN
Dataset – Sankalp_Sase
Parameter – Relative Humidity
Duration - 3 years

LSTM

Long Short Term Memory

Introduction to LSTM:

- LSTM is a type of a Recursive Neural Network.
- LSTM can be thought of as an improvement over RNN Models.
- Two common problems associated with deep RNN models are: 1. The problem of exploding gradients and 2. The problem of vanishing gradients.
- The LSTM model deals with these problems by the usage of Gates.

Structure:

Each unit of an LSTM network has four parts:

- The Memory Cell (the learned weight)
- The Input Gate
- The Output Gate
- The Forget Gate

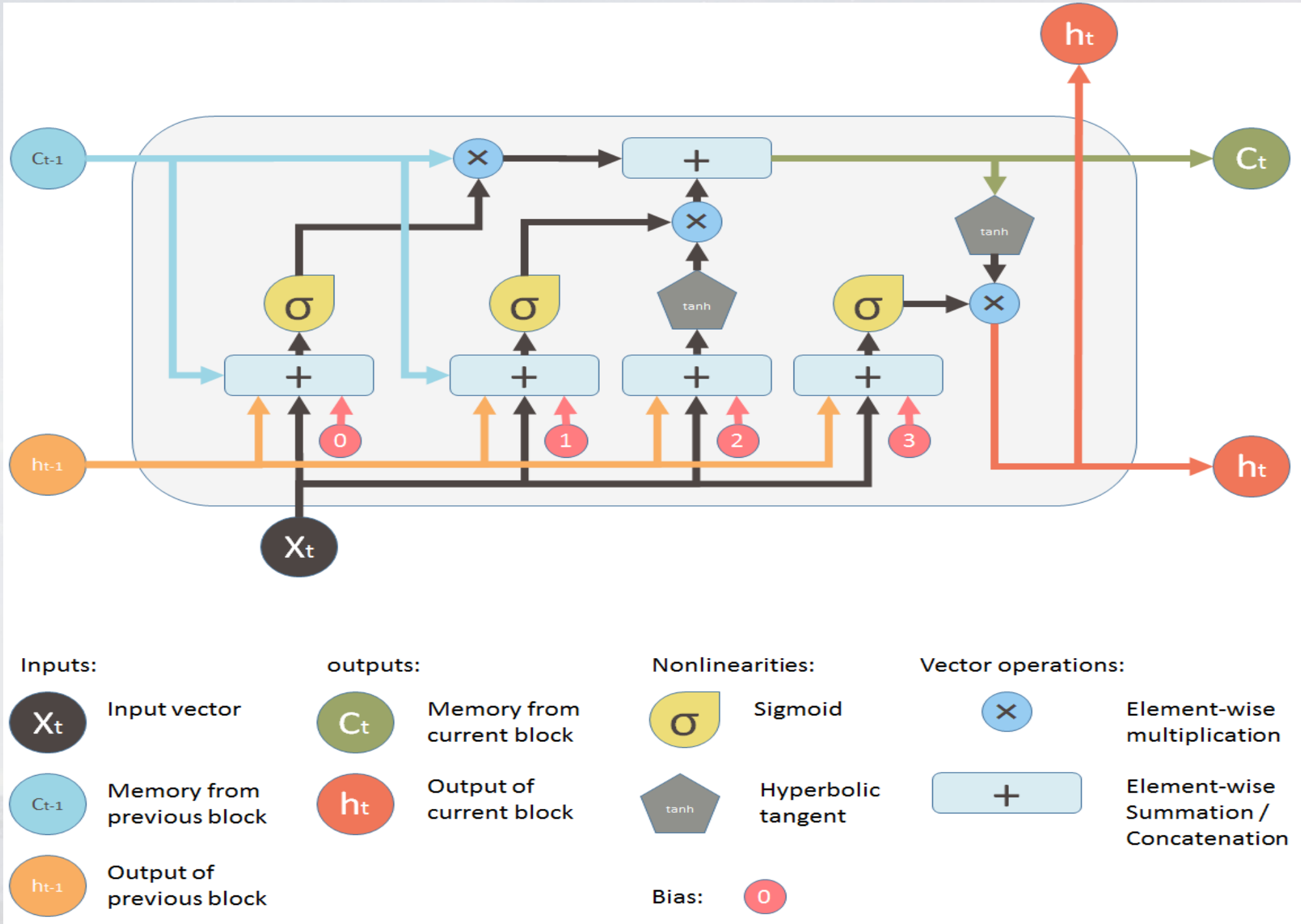
Functionality:

At each time step, each of the gates affects the value of the memory cell or the output:

- The Input Gate decides which portion of the previous memory cell to include in the current time step.
- The Output Gate decides which portion of the current Time Step to include in the subsequent time steps.
- The Forget Gate decides which portion of the Memory Cell to forget.

Dynamicity:

The LSTM model used in the project is Dynamic in Nature. This means that it will work for any dataset whose forecast is needed without the need of a Data Scientist to optimize it every time.


ht
Ct-1
Ct
tanh
σ
σ
tanh
σ
×
+
+
+
+
0
1
2
3
ht-1
ht
Xt
Inputs:
Input vector
Xt
Memory from previous block
Ct-1
Output of previous block
ht-1
outputs:
Memory from current block
Ct
Output of current block
ht
Nonlinearities:
Sigmoid
σ
Hyperbolic tangent
tanh
Vector operations:
Element-wise multiplication
×
Element-wise Summation / Concatenation
+
Bias:
0

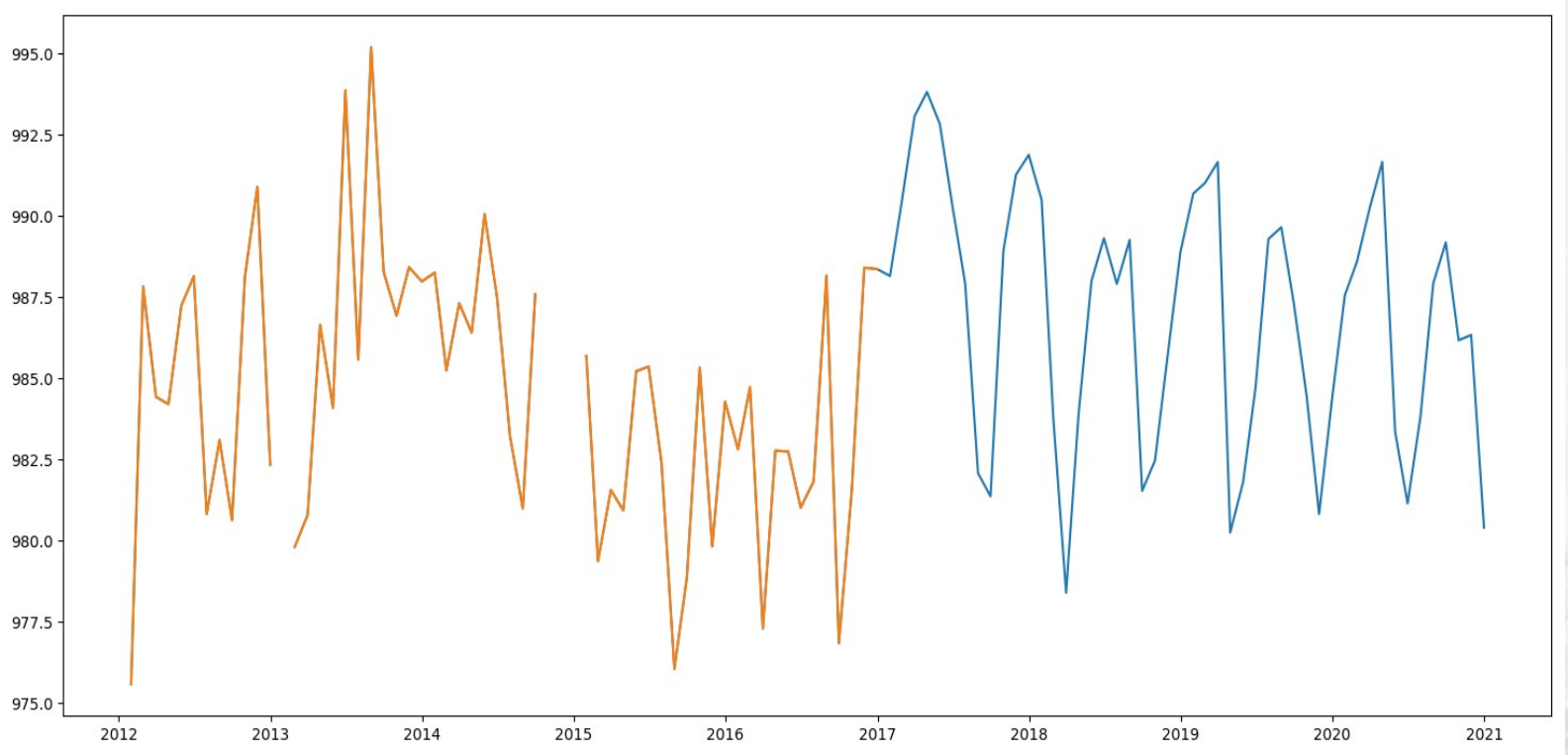
Implementation

The Keras library was used for easily implementing LSTM.

```
model=Sequential()
model.add(LSTM(100,input_shape=(12,1),))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam'
)
ds2=np.reshape(ds2, (ds2.shape[0], 1, 1))i=0
while i==0 or hist.history['loss'][0]>0.02:
   i+=1
   print('Epoch: ',i)
   hist=model.fit(X, Y, epochs=1, verbose=1,
batch_size=1)
```

Result

LSTM gives the best prediction out of all the models if the data is sufficiently large.



Model - LSTM
Dataset - IIG Bharati
Parameter - Air Pressure
Duration - 4 years

ARIMA

Auto Regression
Integration
Moving Average

Introduction to ARIMA:

- It is a Statistical Model.
- It is mainly applied in statistics and econometrics.

Functioning:

The ARIMA model has 3 different parameters p, d and q:

- p is the constant related to Auto Regression.
- d is the Integration Parameter.
- q is the constant corresponding to the Moving Average model.

p - value:

- The parameter p is part of the AutoRegressive model.
- Determined by plotting an AutoCorrelation function.
- In an AR(p) model the future value of a variable is assumed to be a linear combination of p past observations and a random error together with a constant term.

d - value:

- The parameter d is the Integration parameter.
- It makes non-stationary data stationary by applying differencing.
- Eg. First order differencing means that taking the difference between every consecutive value in the time series would make it free from any trend.

q - value:

- The parameter q is part of the Moving Average model.
- It is determined by plotting a Partial AutoCorrelation function.
- MA(q) model uses past errors as the explanatory variables to predict the future outcome.

Implementation

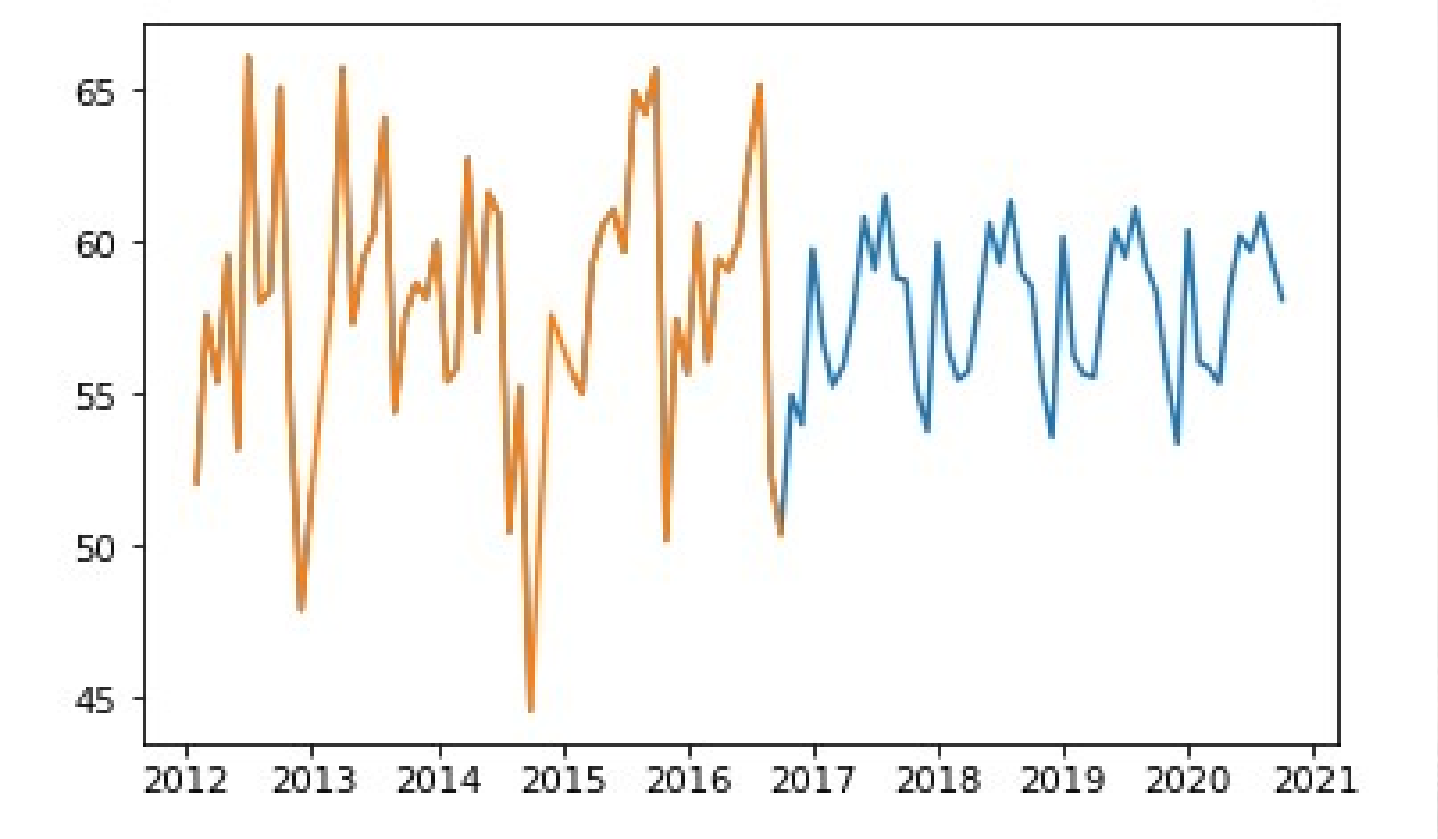
The pmdarima library was used for implementing ARIMA.

```
pipe = pipeline.Pipeline(
[(
"fourier", ppc.FourierFeaturizer(m=12)),
("arima",arima.AutoARIMA(stepwise=True, trace=1,
error_action="ignore", seasonal=False, params=False,
suppress_warnings=True))
])

    pipe.fit(data)
```

Result

LSTM gives the best prediction out of all the models if the data is sufficiently large.



Model - ARIMA
Dataset - IIG Bharati
Parameter - Relative Humidity
Duration - 4 years

Follow - up:

- These models can be accessed on the webpage : http://data.ncaor.gov.in/
- All the mentioned datasets and parameters will be available with all the models at the end of the project.


THANK YOU