RV College of Engineering®

Department of
Electrical & Electronics Engineering

Go, change the world

Minor Project

Deep Learning models on Time-series data of Solar Power for Forecasting applications.

Group B19
Under the Guidance of,
Prof.Sushmita Sarkar

Sai Aslesh Bollepalli	**1RV18EE046**
Vaibhav Jha	**1RV18EE059**
Vedant Chandak	**1RV18EE060**

TABLE OF CONTENTS

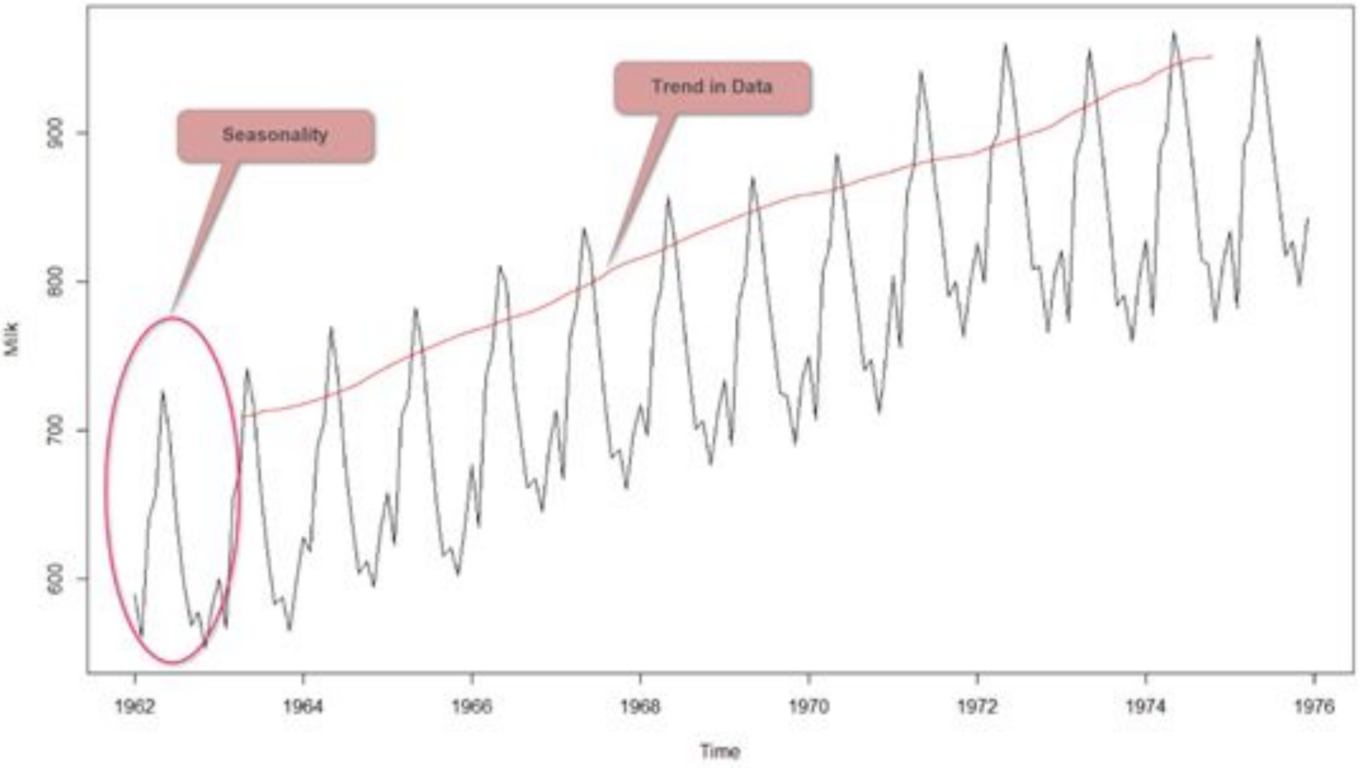
INTRODUCTION

Time series can be defined as a sequence of a metric that is recorded over regular time intervals. Depending on the frequency, a time series can be yearly, quarterly, monthly, etc

Components of time series data:

- Trend
- Seasonality
- Cyclical component
- Noise (error)

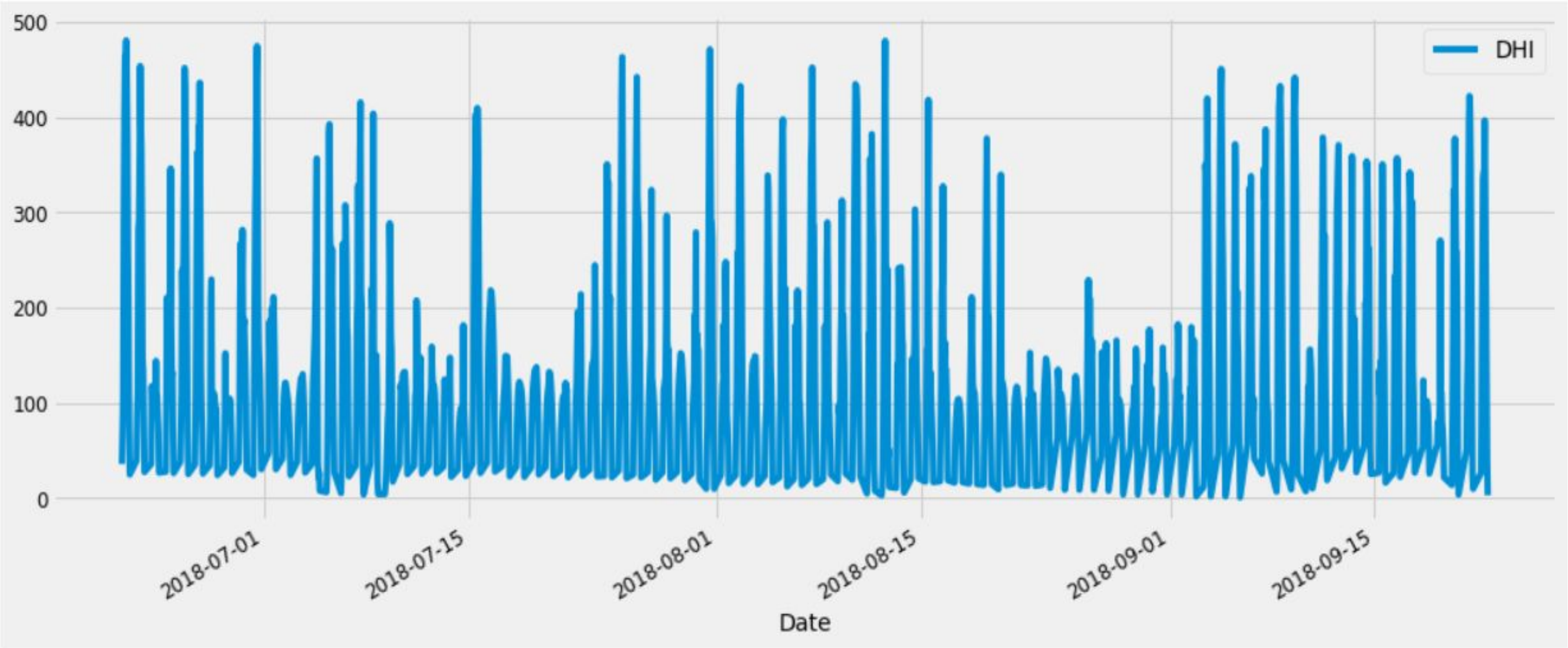
INTRODUCTION



DATA VISUALIZATION

1	Source	Location ID	City	State	Country	Latitude	Longitude	Time Zone	Elevation	Local Time Zone	Clearsky DHI Units	Clearsky DNI Units	Clearsky GHI Units	Dew Point Units	DHI Units
2	NSRDB	649664	-	-	-	29.53	-98.78	-6	327	-6	w/m2	w/m2	w/m2	c	w/m2
3	Year	Month	Day	Hour	Minute	DHI	DNI	GHI	Dew Point	Surface Albedo	Wind Speed	Wind Direction	Relative Humidity	Temperature	Pressure
4	2018	1	1	0	0	0	0	0	-9.4	0.11	5.7	17	58.63	-2.4	996
5	2018	1	1	0	30	0	0	0	-9.4	0.11	5.5	17	59.5	-2.6	996
6	2018	1	1	1	0	0	0	0	-10	0.11	5.4	17	57.79	-2.8	996
7	2018	1	1	1	30	0	0	0	-10	0.11	5.3	18	58.65	-3	996
8	2018	1	1	2	0	0	0	0	-10.5	0.11	5.2	19	57.31	-3.2	996
9	2018	1	1	2	30	0	0	0	-10.5	0.11	5.1	18	57.74	-3.3	996
10	2018	1	1	3	0	0	0	0	-10.8	0.11	5.1	18	56.44	-3.4	996
11	2018	1	1	3	30	0	0	0	-10.8	0.11	5.1	17	56.44	-3.4	996
12	2018	1	1	4	0	0	0	0	-11.2	0.11	5	17	55.24	-3.5	997

DATA VISUALIZATION



LITERATURE SURVEY

Author	Paper/Book Title	Publication Details	Summary	Remark
Jianqin Zhenga , Haoran Zhangb, , Yuanhao Daia , Bohong Wanga , Taicheng Zhenga , Qi Liaoa , Yongtu Lianga , Fengwei Zhangc , Xuan Song	Time series prediction for output of multi-region solar power plants [1]	Applied Energy Volume 257, 1 January 2020, 114001,Elsevier	• Solar power output prediction, it can be a decision-making tool in power system operations. • algorithm is applied to optimize the parameters of the long short-term memory model, and the sensitivity of the different divisions of datasets and the setting of parameters	• solar power generation is a fluctuating power source that is heavily reliant on weather conditions, resulting in uncertainty and intermittency of solar energy
Rich H. Inman, Hugo T.C. Pedro, Carlos F.M. Coimbra	Solar forecasting methods for renewable energy integration[2]	Progress in Energy and Combustion Science Volume 39, Issue 6, December 2013, Pages 535-576,Elsevier	• Reviews some fundamental concepts and methods • numerous approaches presented, it is clear that some level of predictive success • Models are compared, both long term and short-term models	• Solar power output depends on solar irradiance components, airmass, Linke turbidity, clear sky models, clear sky and clearness indices • irradiance observations tend to work well in both data-poor and data-rich environments • Artificial Neural Network (ANN) modeling offers improved nonlinear approximator performance

LITERATURE SURVEY

Author	Paper/Book Title	Publication Details	Summary	Remark
Mohamed Lotfi, Mohammed Javadi, Gerardo J. Osório, Cláudio Monteiro and João P. S. Catalão	A Novel Ensemble Algorithm for Solar Power Forecasting Based on Kernel Density Estimation[3]	Lotfi, M.; Javadi, M.; Osório, G.J.; Monteiro, C.; Catalão, J.P.S. A Novel Ensemble Algorithm for Solar Power Forecasting Based on Kernel Density Estimation. Energies 2020, 13, 216.r	• Forecasting is performed by calculating a KDE-based similarity index to determine a set of most similar cases from the historical dataset. Then, the outputs of the most similar cases are used to calculate an ensemble prediction • The algorithm predicts uncertainties associated with high frequency weather variations.	• Ensemble methods are highly successful in forecasting when compared to single Machine Learning and Deep learning algorithms • Simple ensemble methods outperform complex Machine Learning or Artificial Neural Network models.
Cheng Pan, Jie Tan	Day-Ahead Hourly Forecasting of Solar Generation Based on Cluster Analysis and Ensemble Model[4]	C. Pan and J. Tan, "Day-Ahead Hourly Forecasting of Solar Generation Based on Cluster Analysis and Ensemble Model," in IEEE Access, vol. 7, pp. 112921-112930, 2019, doi: 10.1109/ACCESS.2019.2935273.	• Clustering of data according to weather regime. • Determining the weights through ridge regression • Using ensemble model called Random Forest to make the forecast	• The improvement due to usage of ensemble models was stressed again. • To reduce bias due to manual weights, intelligent use of ridge regression was used to calculate weights. • Very important to cluster data according to seasons to improve efficiency of the model.

LITERATURE SURVEY

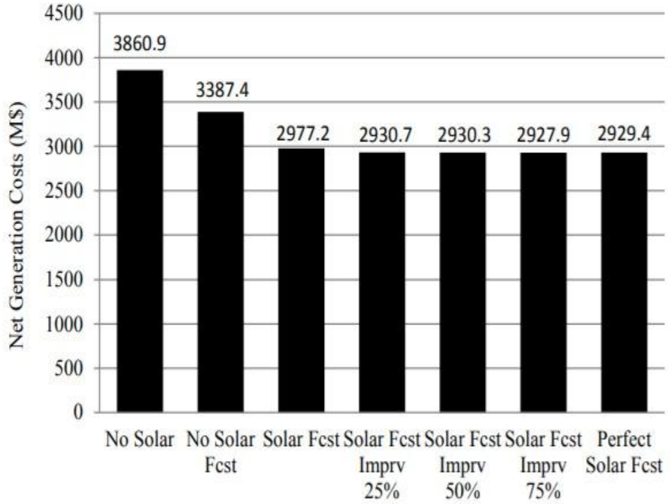
Author	Paper/Book Title	Publication Details	Summary	Remark
Waleed I. Hameed, Baha A. Sawadi, Safa J. Al-Kamil, Mohammed S. Al-Radhi, Raed A. Abd-Alhameed, Ameer L. Saleh, Yasir I. A. Al-Yasir.	Prediction of Solar Irradiance Based on Artificial Neural Networks[5]	• Licensee MDPI, Basel, Switzerland. This article is an open access • article distributed under the terms and conditions of the Creative Commons Attribution • (CC BY) license	• Tells us about the economical and technical advantages of using a pyranometer to calculate the solar irradiance. • Gives an ANN approach to calculate the solar irradiance using a PV panel of 50 watts and 36 cells to extract the Isc and the Voc of the PV module. • ANN uses a feedforward network and use backpropagation to train it properly • Module temperature (T) and Global Irradiance (G) are taken into consideration and the I-V values are converted to the actual G and T conditions using the mathematical equation in the paper. Simulations are run on MATLAB and then graphs are plotted in order to get results.	-Uses ANN to replace pyranometer to calculate the solar irradiance of a horizontal surface. -Using PV panels short circuit current and its open circuit voltage and G(w/m2) for calculation of solar irradiance.

PROBLEM STATEMENT

To develop Deep Learning models on Time-series data of Solar Power for Forecasting applications.

MOTIVATION

1. 25% solar power penetration reduces costs by 22.9%
2. Solar power forecasts are uniformly improved by 25%, the net generation costs are further reduced by 1.56% ($46.5 M).
3. If sunlight energy striking texas is equivalent 300 times the total power output of all the power plants in the world.
4. One of the major issues faced by solar power plants, is for them to maintain long term efficiency of a solar panel and to predict it power output.

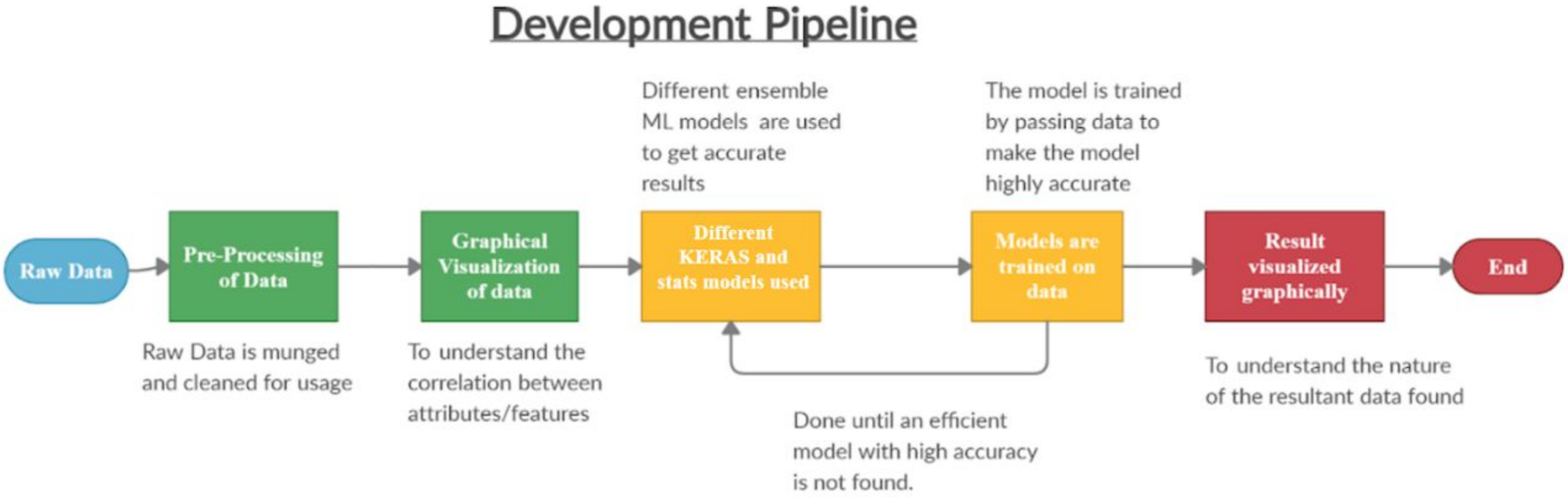


[7]

MAIN OBJECTIVES

1. Collect industry grade data and design robust functions of data wrangling.
   a. NSRDB [8], Solcast [9].
2. To study, review & implement ML models for weather forecasting and work on improving its efficiency.
   a. VAR, ARIMA & Prophet Models.
3. To visualise past and future data and analyse the accuracy of each model
4. To design and implement an efficient and accurate Deep Learning(DL) models, that forecasts solar irradiance.

METHODOLOGY OUTLINE



METHODOLOGY OUTLINE

Stage 1 - Collection of raw, relevant data: NSRDB, NASA power

Stage 2 - Data will be cleansed and munged using OOPs concepts and python libraries like Pandas and NumPy

Stage 3 - Model derives linear/non-linear correlations between data points. Data is visualised. Feature engineering optimizes the results.

METHODOLOGY OUTLINE

Stage 4 - Training ML/DL models on training set and evaluating the model through performance metrics:

a. Mean Absolute Error
b. Root Mean Squared Error
c. Mean Squared Error
d. Mean Absolute Percentage Error

PROGRESS

Software/Tools used :

1. Google Colab - online collaborative code editor
2. Python Libraries - Pandas, Numpy, Datetime, Matplotlib, statsmodel (VAR, ADfuller, RMSE, AIC, coint_johansen, grangercausalitytests, durbin_watson, ARIMA, sm, tsaplots), pmdarima, fbprophet, warnings, itertools.

Models implemented:

1. VAR Model
2. ARIMA Model
3. FB PROPHET Model
4. LSTM (Deep Learning Model)

Data:

1. NSRDB [1]
2. Solcast [2]

PROPHET MODEL

1. Prophet is a univariate time series forecasting model based on **generalised additive model** (GAM) where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
2. Prerequisites for implementing prophet:
   a. Univariate & Stationary time series data.
   b. Datetime and variable must be named ‘ds’ & ‘y’ respectively.
3. The prophet model is generally defined as:

   $y(t) = g(t) + s(t) + h(t) + e(t)$

PROPHET

MATHEMATICS OF PROPHET

General function is defined as: **$y(t) = g(t) + s(t) + h(t) + e(t)$**

where:

- $g(t)$: **trend**
- $s(t)$: **seasonality**
- $h(t)$: **holidays**
- $e(t)$: **error term**

1. Prophet is based on GAM and uses “back-fitting” to determine the functions that fit the data.
2. Back-fitting is an algorithm similar to Gauss-Seidel method for solving linear equations.

MATHEMATICS OF PROPHET

Detailed look at components of y(t): **y(t) = g(t) + s(t) + h(t) + e(t)**

g(t): Trend

1. Package offers 2 trend models "a saturating growth model, and a piecewise linear model."
2. Linear model was chosen due to no absolute saturation point for the give problems statement.

$$g(t) = (k + a(t)^T\boldsymbol{\delta})t + (m + a(t)^T\gamma)$$

where:

k : growth rate

$\boldsymbol{\delta}$: rate adjustments

m : offset parameter

MATHEMATICS OF PROPHET

Detailed look at components of y(t): **y(t) = g(t) + s(t) + h(t) + e(t)**

s(t): Seasonality

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

where:

N : Number of data points

P : Period

MATHEMATICS OF PROPHET

Detailed look at components of y(t): **y(t) = g(t) + s(t) + h(t) + e(t)**

h(t) : holidays

1. No tuning required. Holidays have no effect on solar irradiance.

e(t) : error term

1. Error terms are modelled on data points which show unexpected and erratic behaviour.

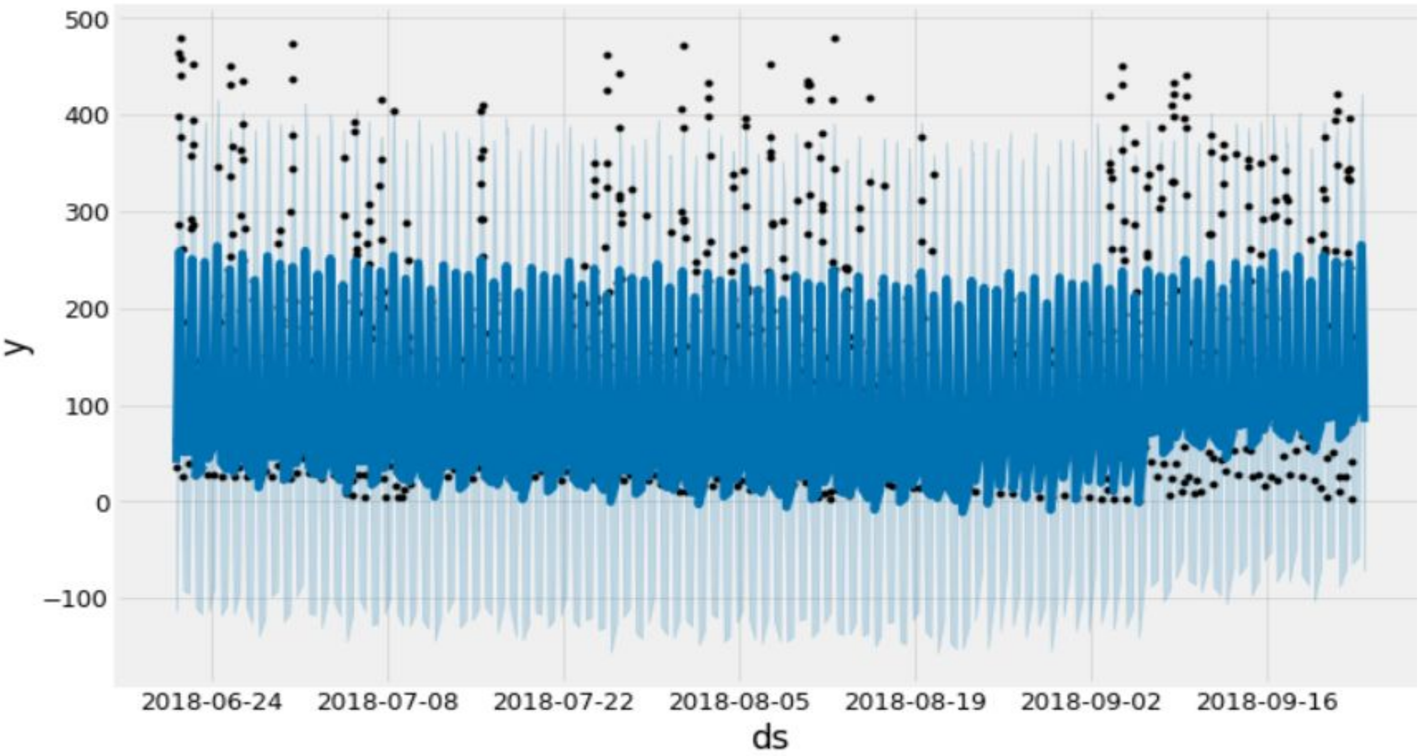
IMPLEMENTING PROPHET

1. Augment the data set according to prophet format.
2. Cleaning the data and Testing for stationarity.
3. Visualising and analysing data.
4. Determining uncertainty interval.
5. Fitting of model to data.
6. Populating a prediction data set and forecasting.
7. Performance metrics and Cross Validation with test set.
8. Tuning the model.

RV College of Engineering®

Go, change the world

PREDICTION ON HISTORICAL DATA



Legend :

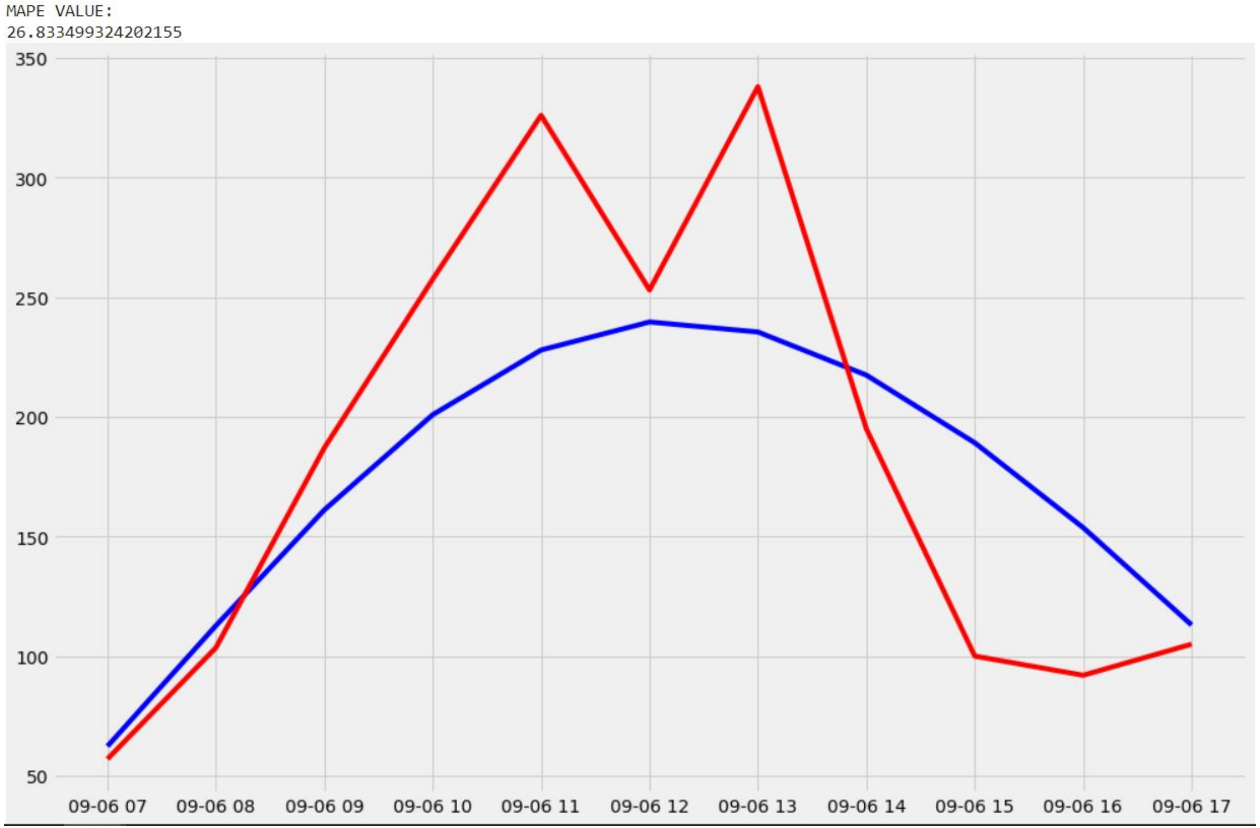
DHI_Forecast

DHI_Actual

X-axis: Date

Y-axis: DHI

FORECAST



legend

Actual value

Forecasted value

MAPE VALUE:
26.8335 %

ARIMA MODEL

1. **Autoregressive Integrated Moving Average (ARIMA)** - is a univariate time series forecasting model, which learns on its own past values, to forecast future values.
2. Basic requirements in order to use ARIMA are:
   - A stationary time series univariate data.
3. Arima model is characterised by three terms.
   1) p - The order of the Autoregressive term.
   2) q - The order of the Moving Average term.
   2) d - The number of differencing required to make the time series stationary.

MATHEMATICS OF ARIMA

1. **AutoRegression (AR) model equation**

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

2. **Moving Average (MA) model equation**

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

3. **Errors Et and E(t-1) equation (**The error equation of respective lags**)**

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_0 Y_0 + \epsilon_t$$

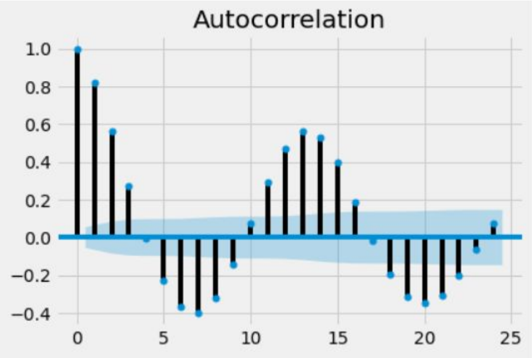
$$Y_{t-1} = \beta_1 Y_{t-2} + \beta_2 Y_{t-3} + .. + \beta_0 Y_0 + \epsilon_{t-1}$$

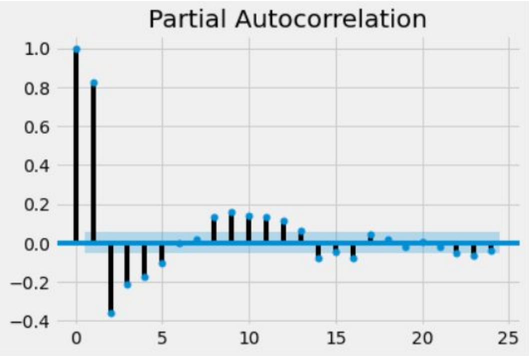
4. **AR and MA model equation (**Upon combining AR and MA equations**)**

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

TESTS FOR ARIMA MODEL

1. ACF and PACF Plots: ACF gives us values of auto-correlation of any series with its lagged values. PACF it finds correlation of the residuals, with the next lag value.
   It is used to cross-validate the order of AR and MA.





2. Stationarity : **Augmented Dickey-Fuller Test (ADF Test),** which gives the optimal **d** value in order to make the data stationary (Passed)

SELECTING ORDER

1. Selecting Order: A stepwise iterative search is performed on the data to get the order with the least **AIC (Akaike information criterion)**

```
Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=13978.604, Time=0.12 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=16766.405, Time=0.03 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=14059.137, Time=0.08 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=15498.357, Time=0.18 sec
 ARIMA(2,0,1)(0,0,0)[0]             : AIC=13954.900, Time=0.26 sec
 ARIMA(2,0,0)(0,0,0)[0]             : AIC=13959.619, Time=0.10 sec
 ARIMA(3,0,1)(0,0,0)[0]             : AIC=13955.369, Time=0.30 sec
 ARIMA(2,0,2)(0,0,0)[0]             : AIC=13954.181, Time=0.38 sec
 ARIMA(1,0,2)(0,0,0)[0]             : AIC=13961.528, Time=0.33 sec
 ARIMA(3,0,2)(0,0,0)[0]             : AIC=13662.546, Time=1.23 sec
 ARIMA(3,0,3)(0,0,0)[0]             : AIC=inf, Time=nan sec
 ARIMA(2,0,3)(0,0,0)[0]             : AIC=13950.805, Time=0.70 sec
 ARIMA(3,0,2)(0,0,0)[0] intercept   : AIC=inf, Time=2.86 sec

Best model:  ARIMA(3,0,2)(0,0,0)[0]
Total fit time: 7.765 seconds
```

- **Best model :** ARIMA(3,0,2)(0,0,0)[0]
- **Least AIC Value :** 13662.546
- **Total Fit time :** 7.765 seconds.

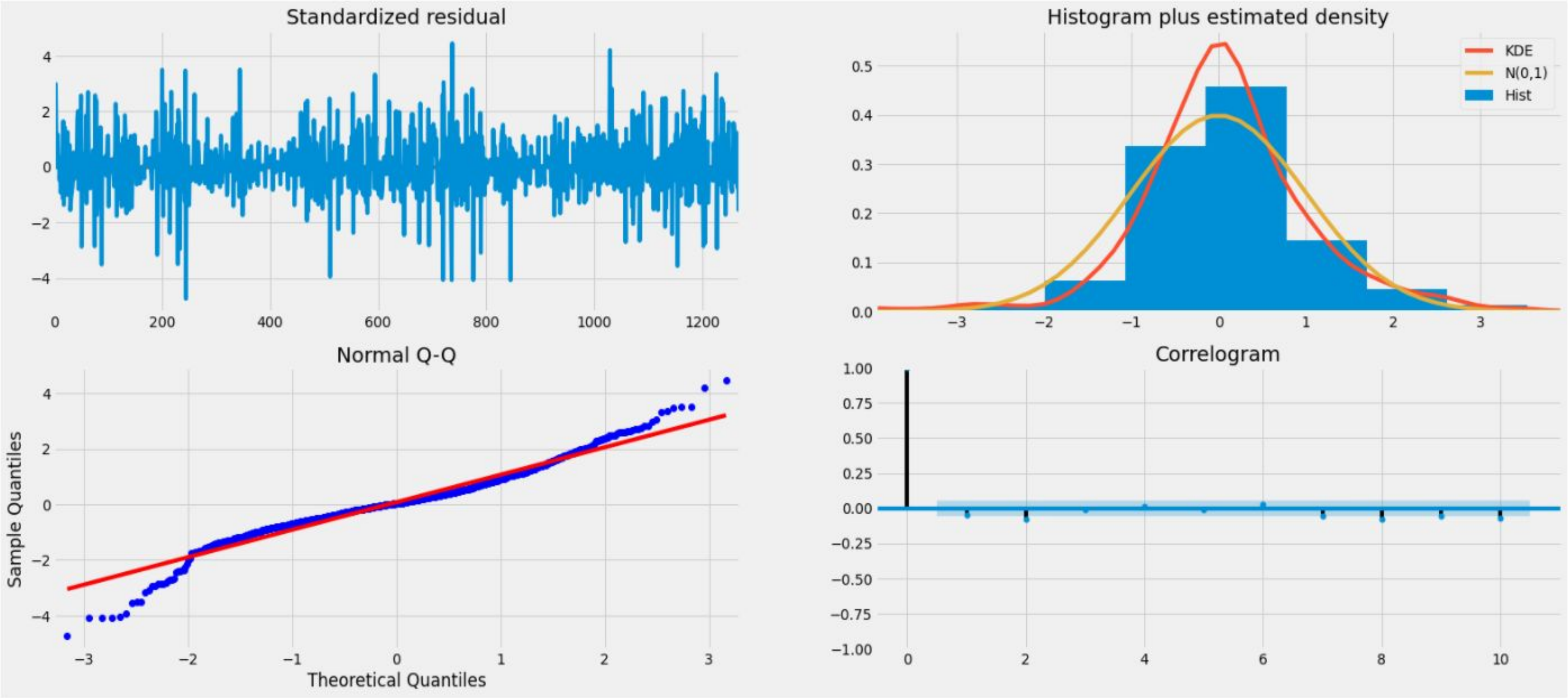
ARIMA MODEL STEPS

1. Munge the data.
2. Select the desired variable column from the dataset.
3. Test for stationarity (ADF test).
4. Run the model to the get the order (p, d, q values using ACF, PACF, AIC calculation).
5. Select the combination of order which gives the least AIC.
6. Plot Normal Q-Q, KDE, Correlogram, Standardised Residual for better analysis.
7. Run the model for that order on the training set.
8. Predict the values for the test sets, and plot it against original test set.
9. Set the periods in future for which we need to get the future forecast.

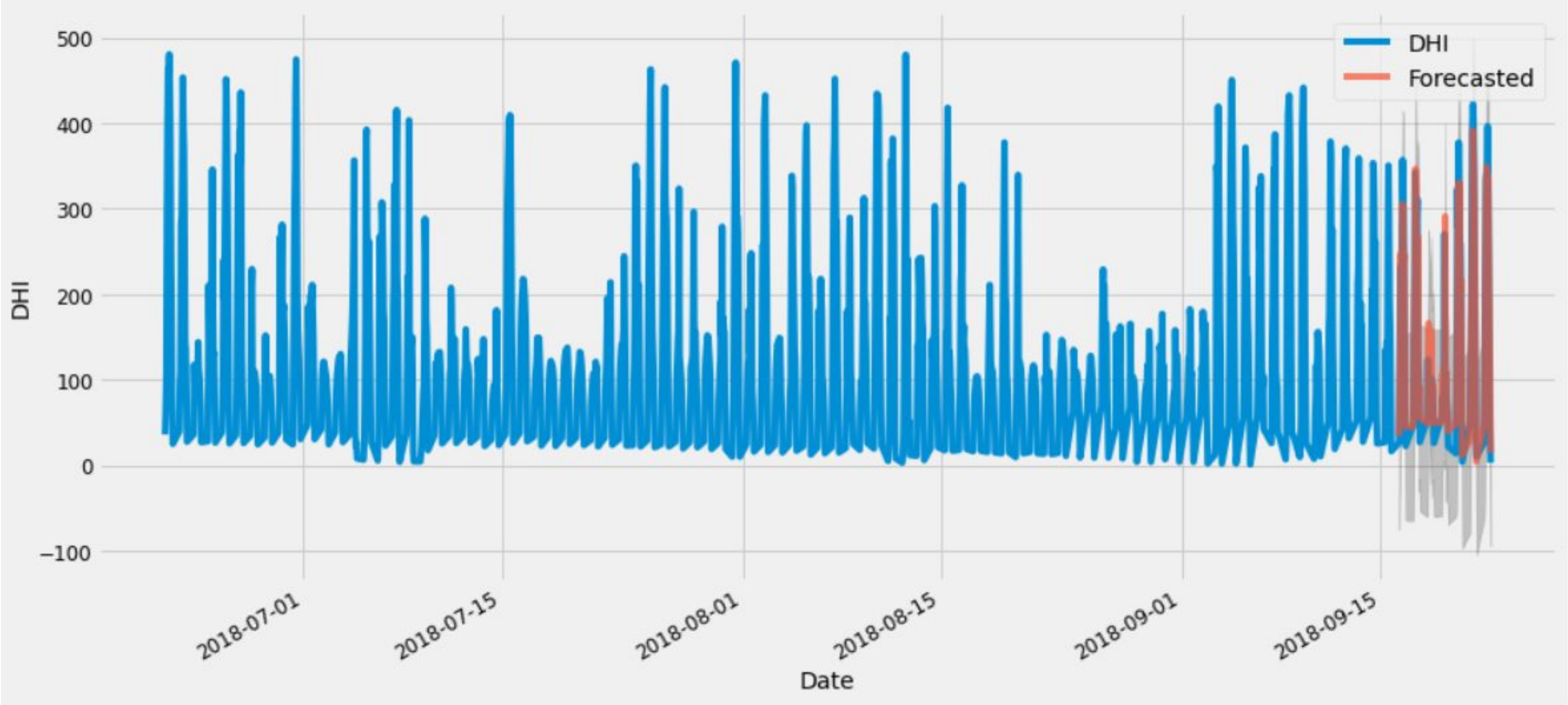
RESULTS - Parameters of the model for the order

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1269
Model:               SARIMAX(3, 0, 2)   Log Likelihood               -6825.273
Date:                Mon, 09 Nov 2020   AIC                          13662.546
Time:                        06:37:29   BIC                          13693.422
Sample:                             0   HQIC                         13674.144
                               - 1269
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          2.7367      0.010    272.754      0.000       2.717       2.756
ar.L2         -2.6737      0.019   -137.727      0.000      -2.712      -2.636
ar.L3          0.9340      0.010     96.012      0.000       0.915       0.953
ma.L1         -1.7408      0.017   -104.148      0.000      -1.774      -1.708
ma.L2          0.8457      0.018     48.284      0.000       0.811       0.880
sigma2      2786.9093     77.780     35.831      0.000    2634.464    2939.355
===================================================================================
Ljung-Box (Q):                      104.54   Jarque-Bera (JB):               458.20
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.97   Skew:                            -0.09
Prob(H) (two-sided):                  0.74   Kurtosis:                         5.94
===================================================================================
```

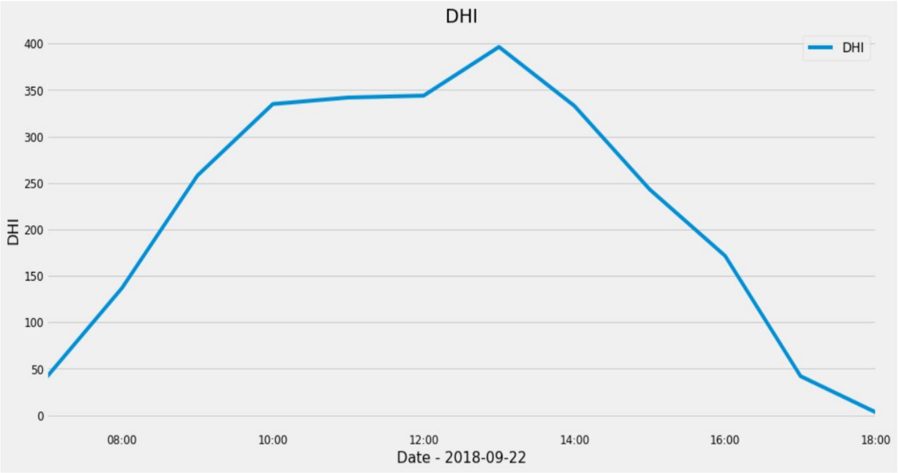
RESULTS - Parameter plots for analysis of model

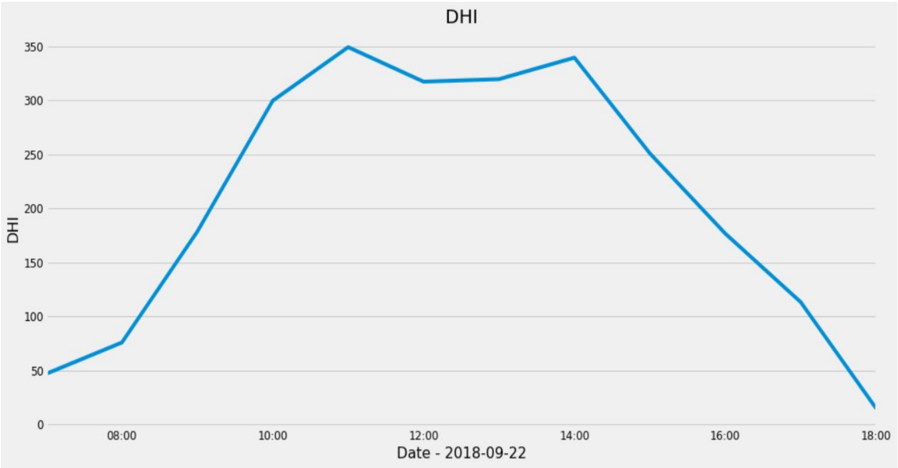


RESULTS - Forecasted test set plot



RESULTS - Real vs Predicted forecast





Real Data Interpretation on 22nd Sept

Predicted Data Interpretation on 22nd Sept

RESULTS - Statistical error analysis

- **MSE -** Mean squared error is found of the forecasted data with actual data to check the accuracy of the forecast
  Value - 3331.8507

- **RMSE -** Similar to MSE, Root Mean Squared Error is found for the same.
  Value - 57.7222

- **MAPE -** Similar to MSE, and RMSE, Mean Absolute Percentage Error is found for the same.
  Value - 43.3842

VAR MODEL

1. **Vector Autoregression (VAR)** is a multivariate forecasting algorithm that is used when two or more time series influence each other.
2. Basic requirements to use VAR are:
   - You need at least two time series (variables).
   - The time series should influence each other.

3. The primary difference between VAR and other AR models, Vector Auto Regression (VAR) is bi-directional i.e., the variables influence each other. Unlike other AR models which are uni-directional.

VAR MODEL STEPS

- **VAR model involves the following steps**:

1. Analyse the time series characteristics
2. Prepare training and test datasets
3. Test for causation (Granger's Causality Test)
4. Test for stationarity (ADF Test)
5. Find optimal order (using AIC Value)
6. Forecasting the values
7. Evaluate the model against test sets
8. Forecast to future

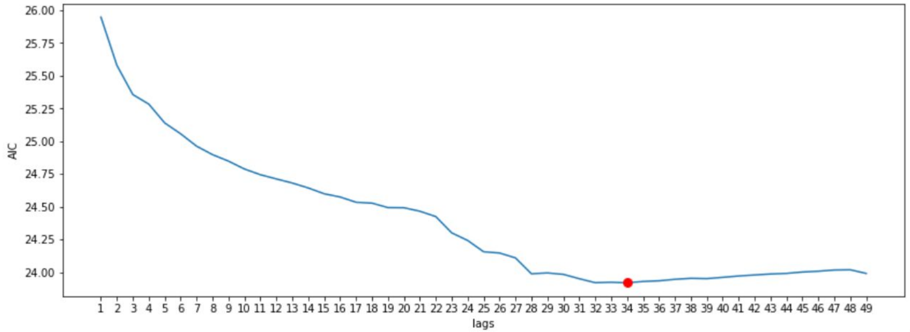
TESTS FOR VAR MODEL AND SELECTING ORDER

1. Testing Causation:Using **Granger's Causality Test**, It's tested if each of the time series in the system influences each other. (Passed)

2. Cointegration Test and Stationarity Test : **Augmented Dickey-Fuller Test (ADF Test)** was used**,** which checks weather all data is stationary (Passed after differencing it once)

3. Selecting Order: The increasing orders of VAR model, were fit iteratively to pick the order that gives a model with least **AIC (Akaike information criterion)**

BEST ORDER 50 BEST AIC: 5.7322



MATHEMATICS OF VAR

1. **AR model equation**

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t$$

2. **VAR model equation (**Two time series variables and first order**)**

$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \epsilon_{2,t}$$

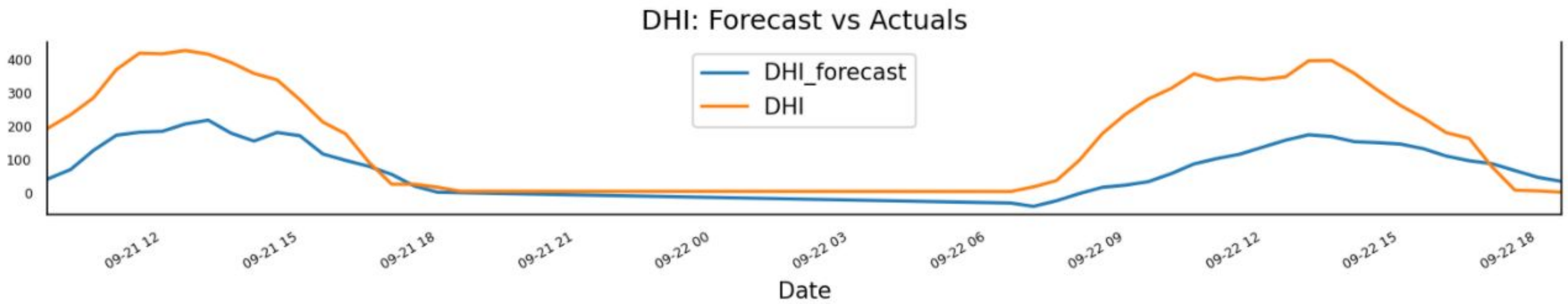
3. **VAR model equation (**Two time series variables and second order**)**

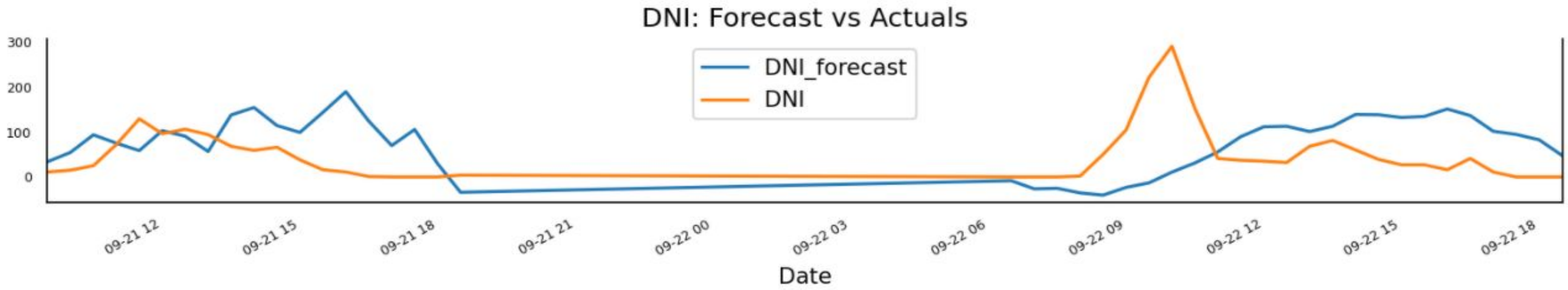
$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \beta_{11,2} Y_{1,t-2} + \beta_{12,2} Y_{2,t-2} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \beta_{21,2} Y_{1,t-2} + \beta_{22,2} Y_{2,t-2} + \epsilon_{2,t}$$

4. **VAR model equation (**Three time series variables and second order**)**

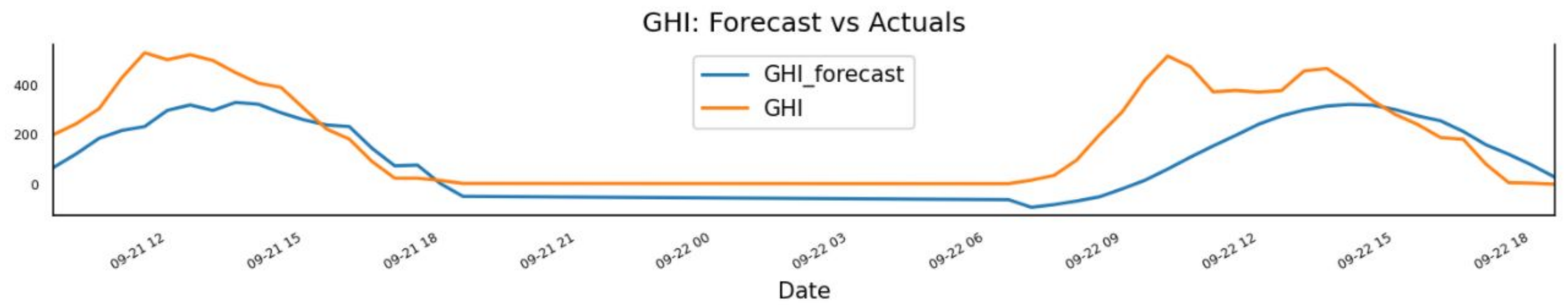
$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \beta_{13,1} Y_{3,t-1} + \beta_{11,2} Y_{1,t-2} + \beta_{12,2} Y_{2,t-2} + \beta_{13,2} Y_{3,t-2} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \beta_{23,1} Y_{3,t-1} + \beta_{21,2} Y_{1,t-2} + \beta_{22,2} Y_{2,t-2} + \beta_{23,2} Y_{3,t-2} + \epsilon_{2,t}$$
$$Y_{3,t} = \alpha_3 + \beta_{31,1} Y_{1,t-1} + \beta_{32,1} Y_{2,t-1} + \beta_{33,1} Y_{3,t-1} + \beta_{31,2} Y_{1,t-2} + \beta_{32,2} Y_{2,t-2} + \beta_{33,2} Y_{3,t-2} + \epsilon_{3,t}$$

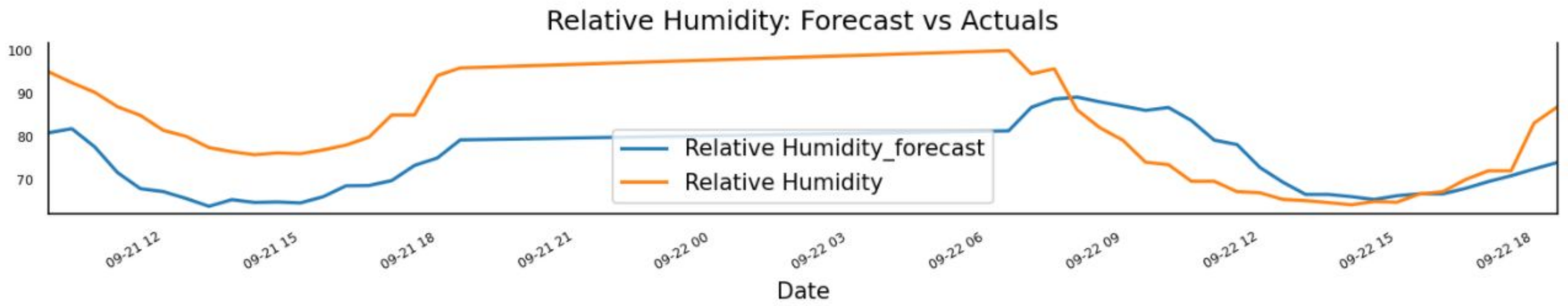
RESULTS VAR MODEL



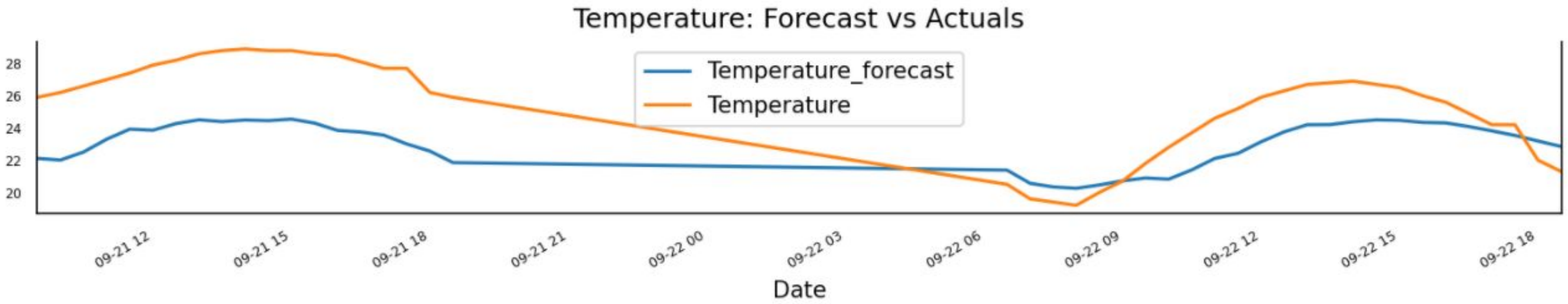


RESULTS VAR MODEL





RESULTS VAR MODEL

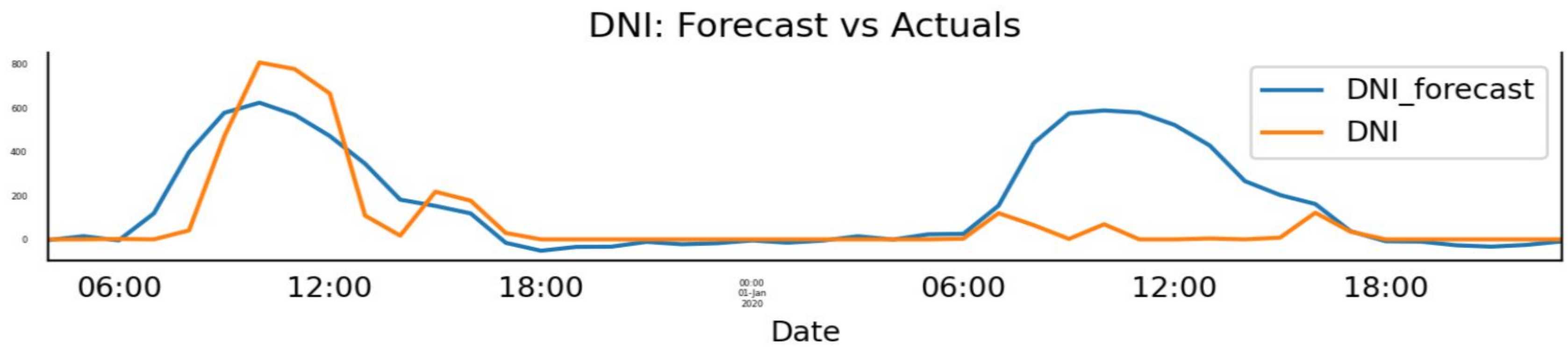


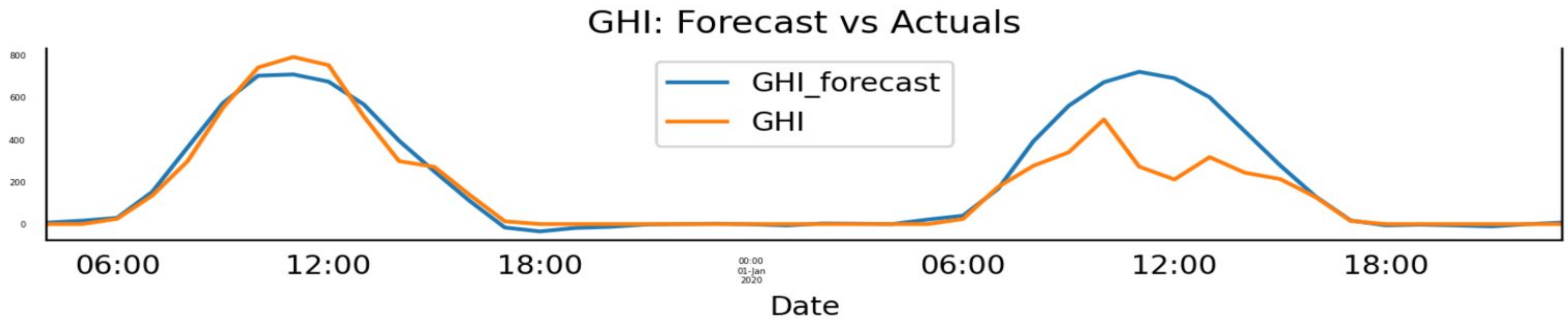
MAPE (Mean Absolute Percentage Error) Values:

MAPE for DHI= 202.62 %
MAPE for GHI= 285.81 %
MAPE for Humidity= 11.45 %
MAPE for Temperature= 10.13 %

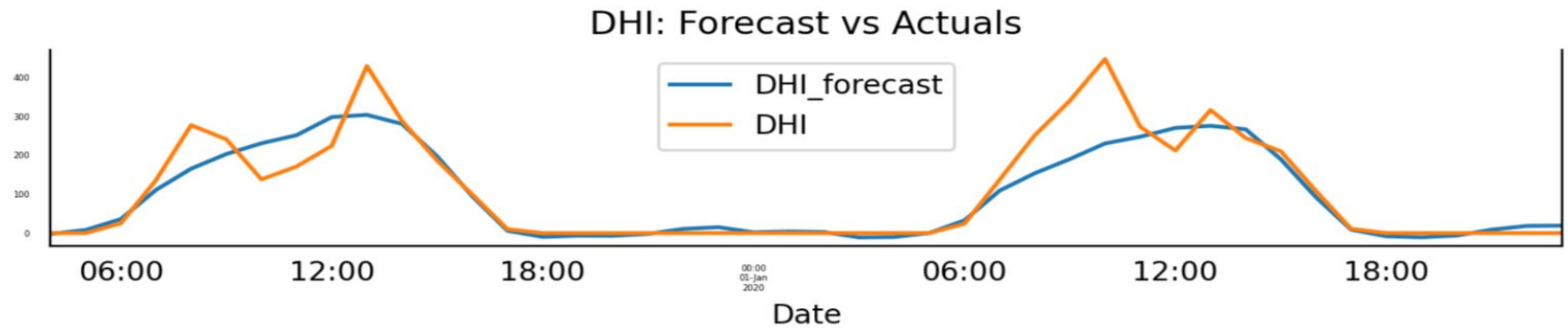
MAPE-value	Accuracy of forecast
Less than 10%	Highly Accurate Forecast
11% to 20%	Good Forecast
21% to 50%	Reasonable Forecast
More than 51%	Inaccurate Forecast

RESULTS VAR MODEL WITH SOLCAST DATA





RESULTS VAR MODEL WITH SOLCAST DATA

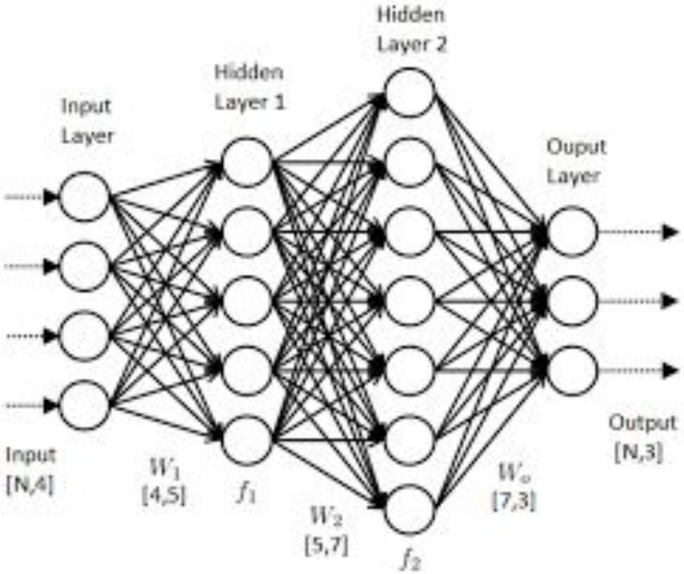



```
The Mean Squared Error of our forecasts is 7218.52
The Root Mean Squared Error of our forecasts is 84.9618
The MAPE of our forecast is 12.1719
```

MAPE-value	Accuracy of forecast
Less than 10%	Highly Accurate Forecast
11% to 20%	Good Forecast
21% to 50%	Reasonable Forecast
More than 51%	Inaccurate Forecast

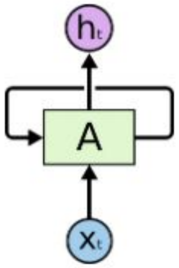
Neural Networks

1. Neural networks are used for approximating relationship(s) between the input & output variable.
2. Universal approximation theorem.

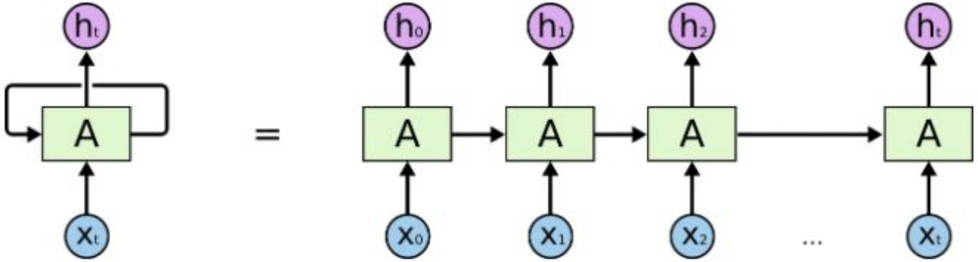


Recurrent Neural Networks

PERSISTENCE



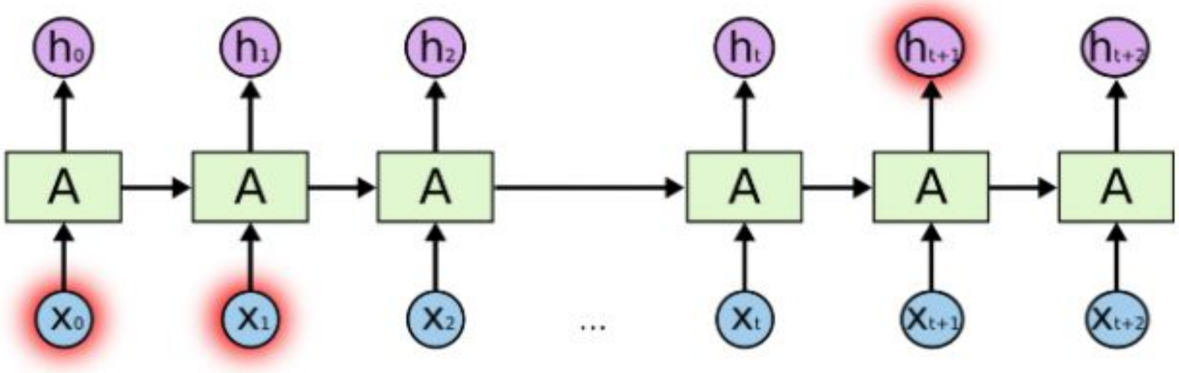
Recurrent Neural Networks have loops.



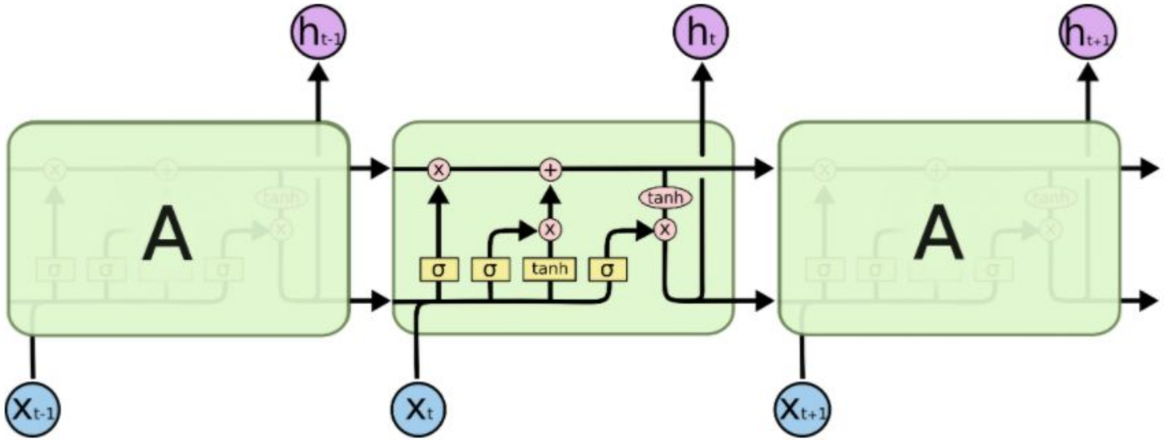
An unrolled recurrent neural network.

Long Term Dependency & LSTM

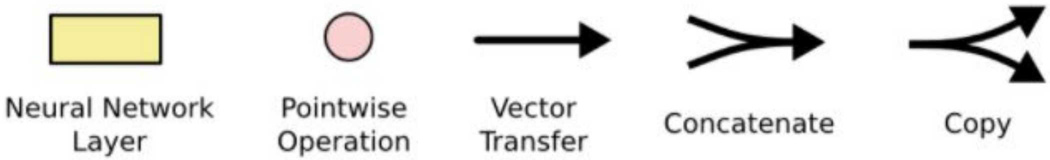
1. RNN suffers from long term dependency.
2. Long Short Term Memory (LSTM) networks.
3. LSTMs are designed to avoid the long term dependency problem.



LSTM (Long Short Term Memory) networks



The repeating module in an LSTM contains four interacting layers.

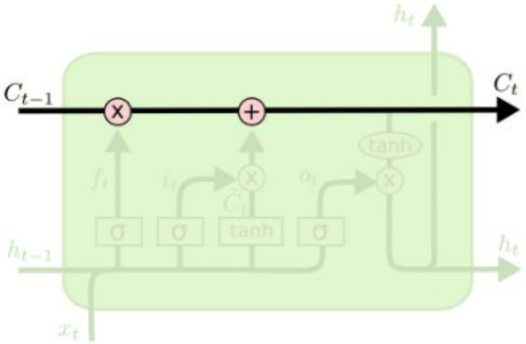


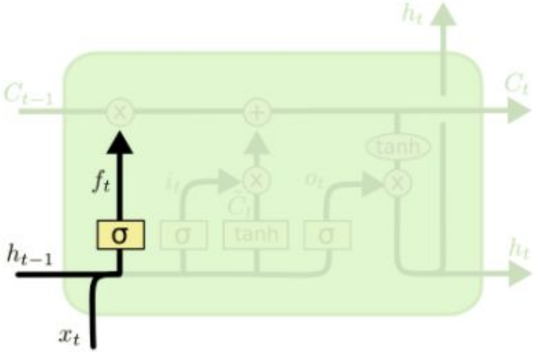
LSTM (Long Short Term Memory) networks

1. Deciding how much of the information to forget.
2. Sigmoid function decides how much of the information to be retained.

   0 : Forget everything.

   1 : Forget nothing

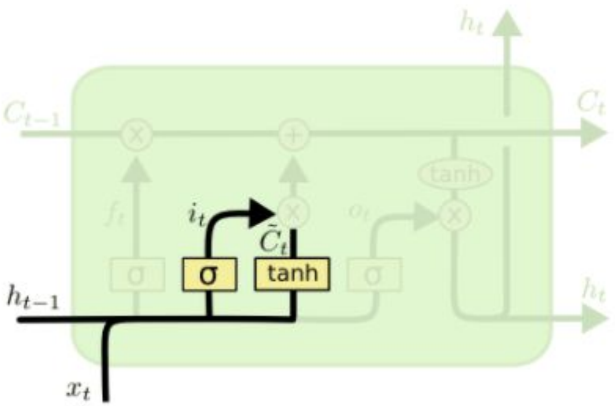




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

LSTM (Long Short Term Memory) networks

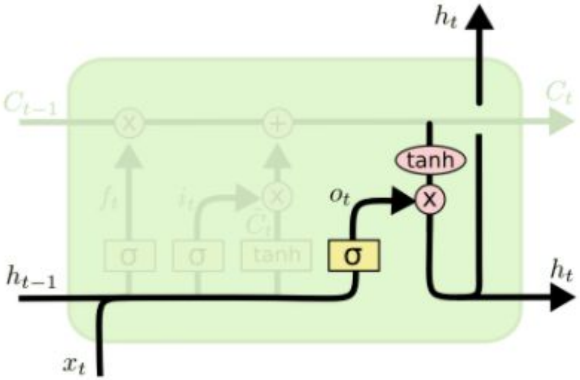
1. Deciding what new information to store in the cell state.
2. A sigmoid layer called the “input gate layer” decides which value(s) to update. Next, a tanh layer creates a vector of new candidate values, that could be added to the state.
3. In the next step, we’ll combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

LSTM (Long Short Term Memory) networks

1. Output will be based on our cell state, but will be a filtered version.
2. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh & multiply it by the output of the sigmoid gate



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Implementation of Multi-Step LSTM

1. Data preparation: Test/Train split
   a. Train : 1 year data of 2018
   b. Test : 1 year data of 2019
2. The LSTM model will maps a sequence of input to an output observation.
3. Divide the sequence into multiple input/output patterns called samples. Using **24** time-steps as input and **24** steps as output.
4. Input shape = **[2161,24,1]**

Implementation of Multi-Step LSTM

Architecture of the LSTM used:

```
[10] #define model
     model = Sequential()
     model.add(LSTM(100, activation = 'relu',return_sequences=True, input_shape=(n_steps_in, n_features)))
     model.add(LSTM(100, activation='relu'))
     model.add(Dense(n_steps_out))
     model.compile(optimizer='adam', loss='mse')
```

```
[11] model.fit(X,y,epochs=50, verbose=0)

     <tensorflow.python.keras.callbacks.History at 0x7f18d0949240>
```

Adam optimiser : Adam optimiser is a modified version of gradient descent.

● Adaptive Gradient Algorithm (AdaGrad) that maintains a per-parameter learning rate that improves performance on problems with sparse gradients.

● Root Mean Square Propagation (RMSProp) that also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight (e.g. how quickly it is changing).

Implementation of Multi-Step LSTM

The shape of the single sample of input data when making the prediction must be **[1,24,1]** for the 1 sample, 24 time steps of the input, and the single feature.

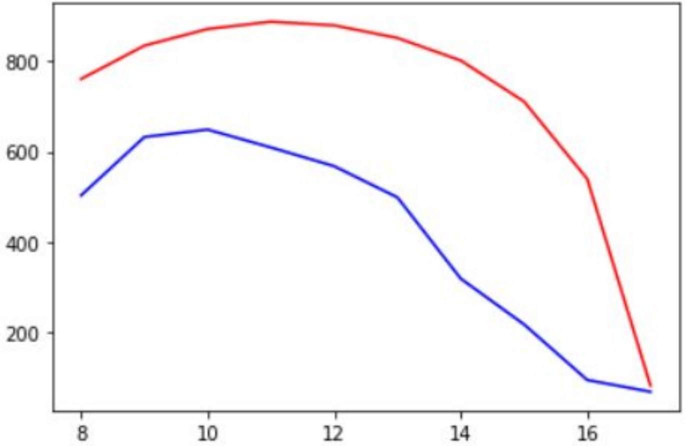
n_steps_in = 24

n_features = 1

```
[40] x_input = np.array(df.iloc[10488:10512,11])
     x_input = x_input.reshape(1,n_steps_in,n_features)
     yhat = model.predict(x_input, verbose=0)
```

Result of Multi-Step LSTM

```
<function matplotlib.pyplot.show>
```



LEGEND

Actual

Prediction

```
[52] mape = np.mean(abs((actual-prediction)/(actual)))*100
```

```
[53] mape
```

```
42.19170038751763
```

APPLICATIONS AND FUTURE SCOPE

1. Solar forecasting provides a way for grid operators to predict and balance energy generation and consumption.
2. The grid operator has a mix of generating assets at their disposal, reliable solar forecasting lets that operator best optimize the way they dispatch their controllable units.
3. Future Scope: Specific focus on Deep learning models and in depth fine tuning of the models for better accuracy.

REFERENCES

1. Jianqin Zhenga , Haoran Zhangb, , Yuanhao Daia , Bohong Wanga , Taicheng Zhenga , Qi Liaoa , Yongtu Lianga , Fengwei Zhangc , Xuan Song “Applied Energy Volume 257, 1 January 2020, 114001,Elsevier "
2. Rich H. Inman, Hugo T.C. Pedro, Carlos F.M. Coimbra “Progress in Energy and Combustion Science Volume 39, Issue 6, December 2013, Pages 535-576,Elsevier.’
3. Mohamed Lotfi, Mohammed Javadi, Gerardo J. Osório, Cláudio Monteiro and João P. S. Catalão “A Novel Ensemble Algorithm for Solar Power Forecasting Based on Kernel Density Estimation”

REFERENCES

4. Cheng Pan, Jie Tan “Day-Ahead Hourly Forecasting of Solar Generation Based on Cluster Analysis and Ensemble Model”
5. Waleed I. Hameed, Baha A. Sawadi, Safa J. Al-Kamil, Mohammed S. Al-Radhi, Raed A. Abd-Alhameed, Ameer L. Saleh, Yasir I. A. Al-Yasir. “Prediction of Solar Irradiance Based on Artificial Neural Networks”
6. Brancucci Martinez-Anido, Carlo & Florita, Anthony & Hodge, Bri-Mathias. (2014). “The Impact of Improved Solar Forecasts on Bulk Power System Operations in ISO-NE”
7. https://ag.tennessee.edu/solar/Pages/What%20Is%20Solar%20Energy/Sun's%20Energy.aspx

REFERENCES

[8] https://nsrdb.nrel.gov/

[9] https://solcast.com/