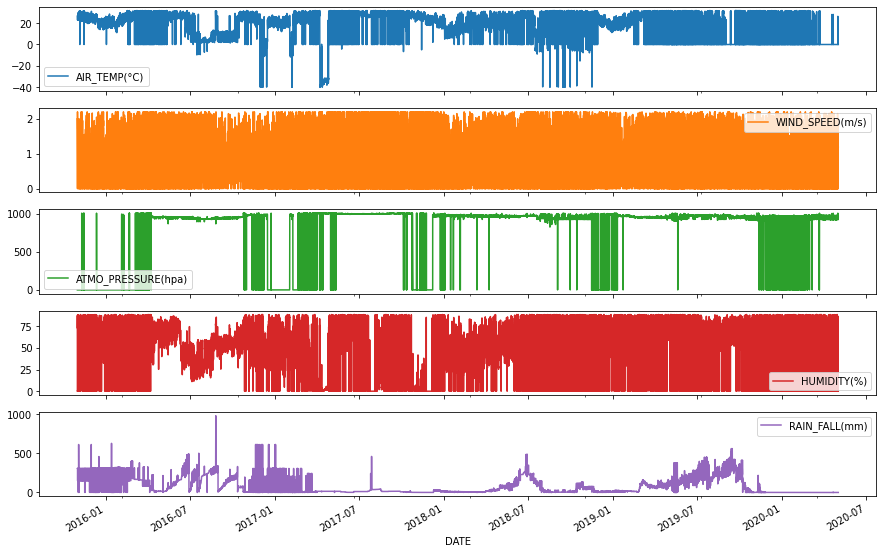
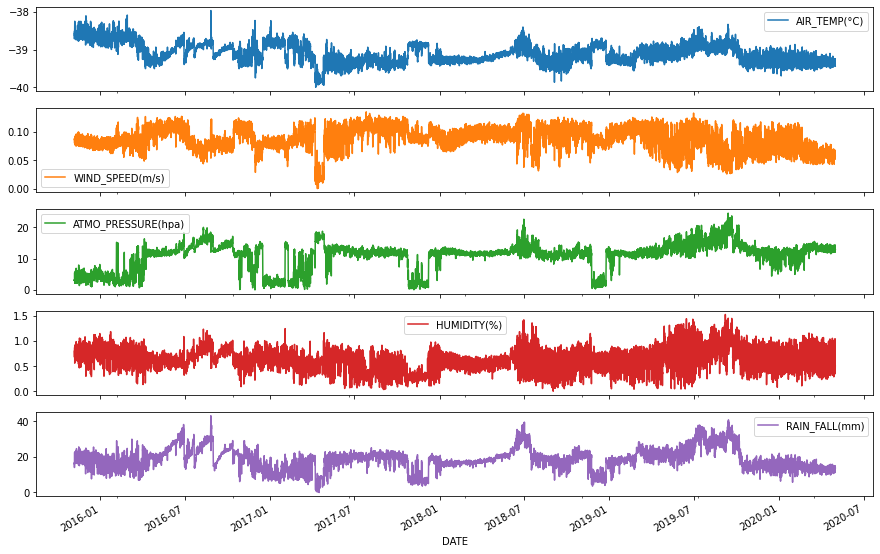
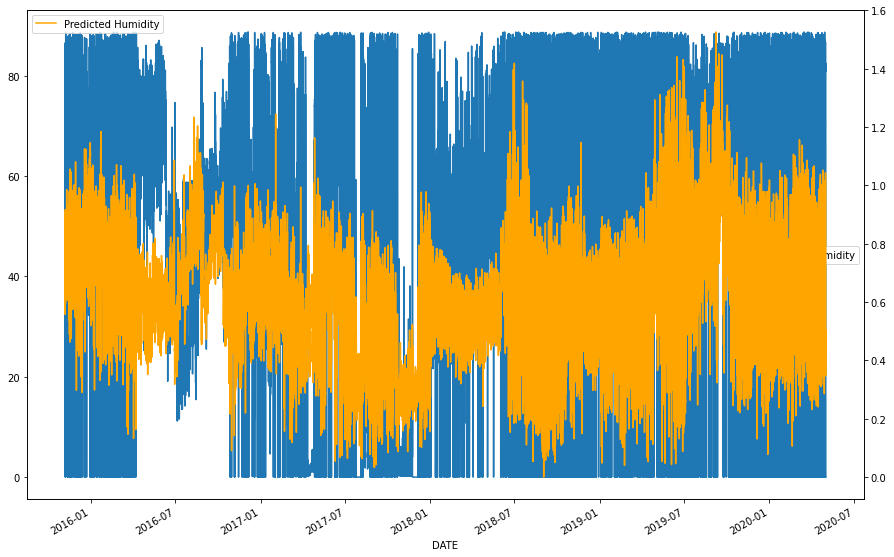
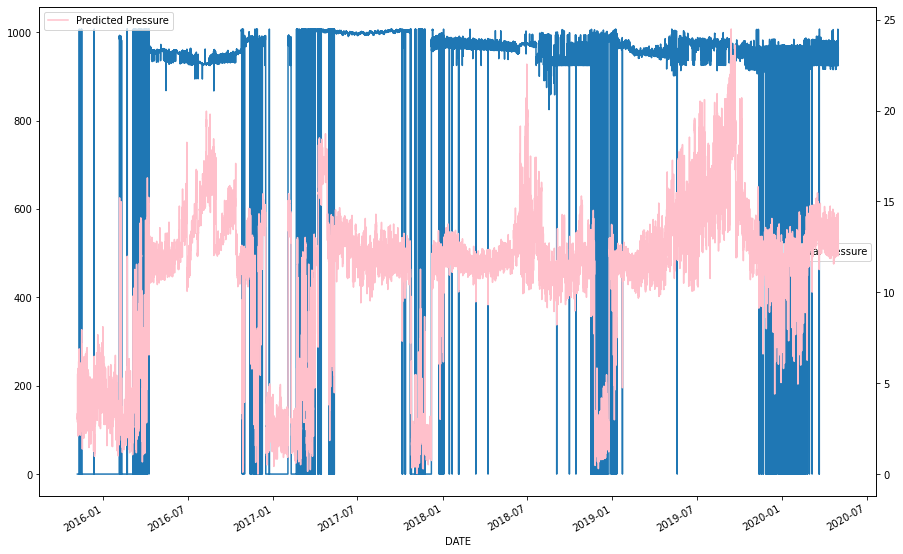
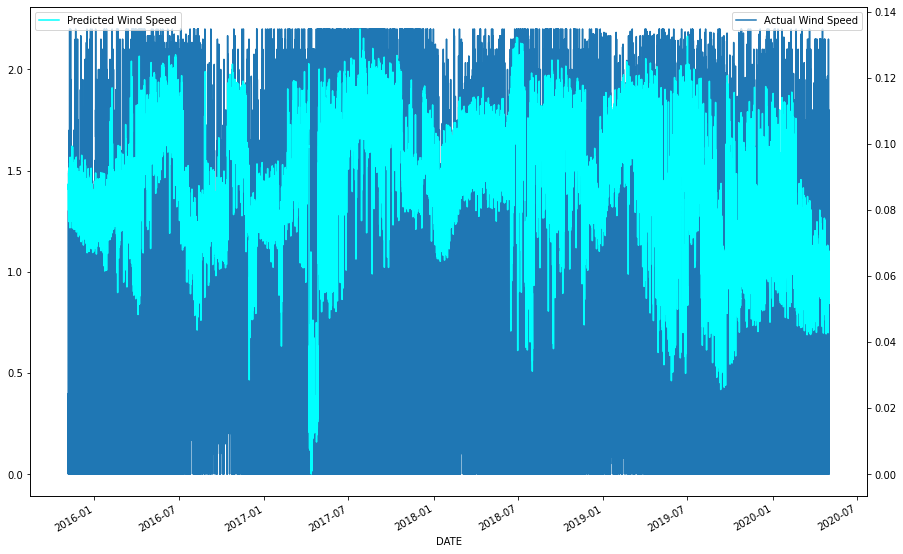
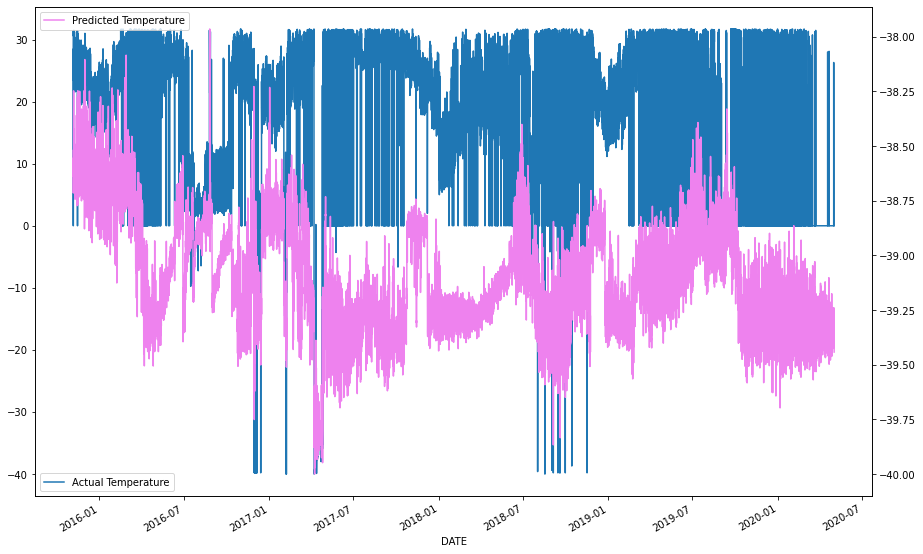
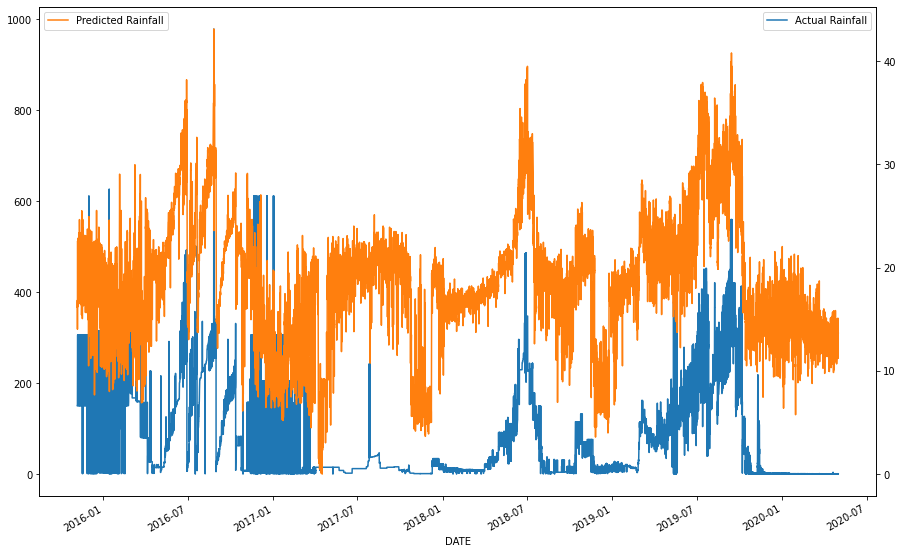
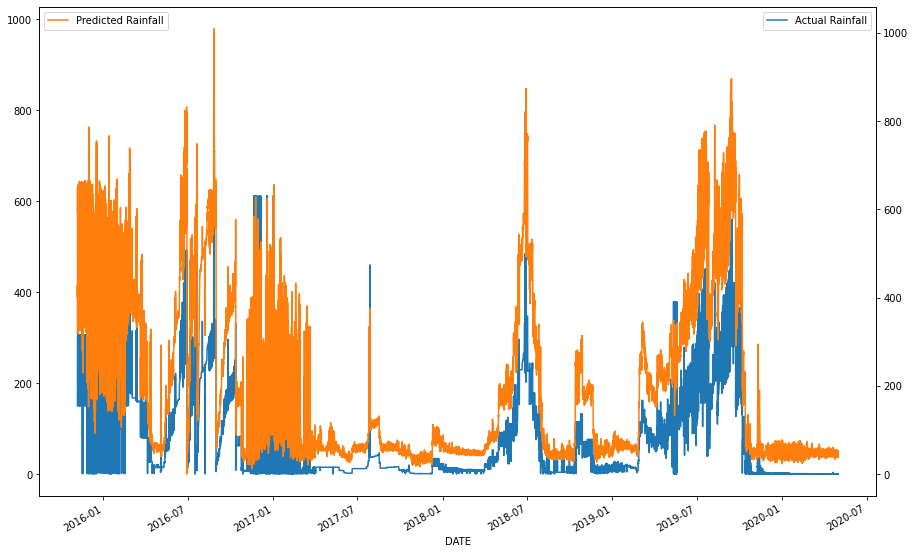
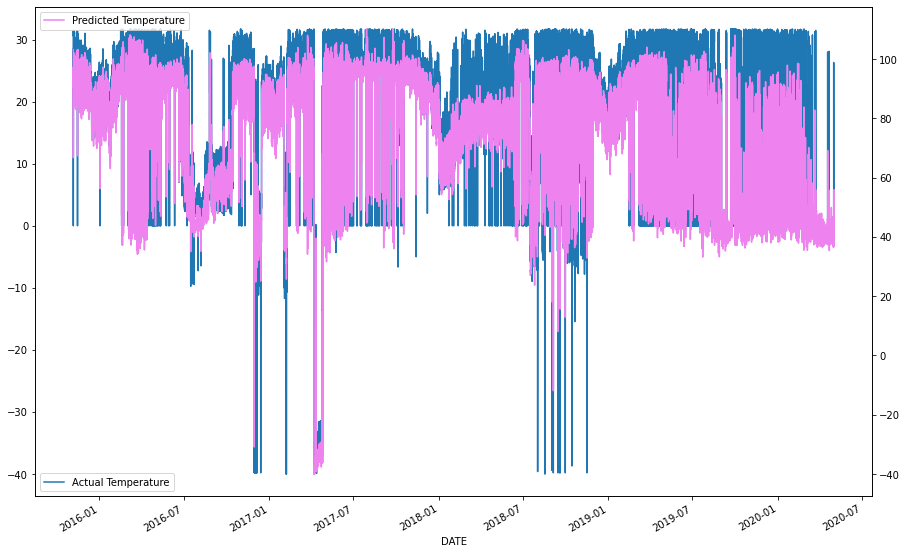
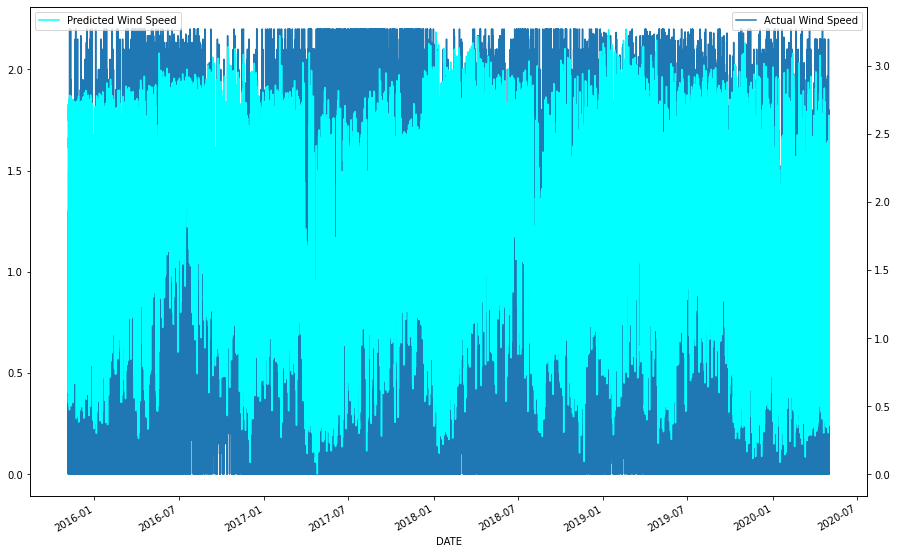
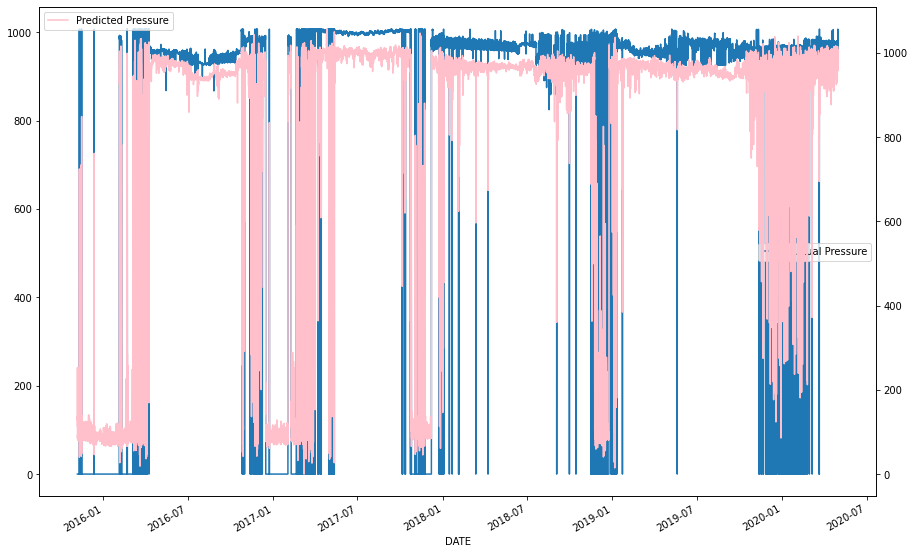


Standardized values

* Multioutput lstm model not possible (not able to predict more than one target features, because inverse scaling would be required for each stem in LSTM model, this kind of work in under research and some methodology can be achieved using convolutional LSTM as described [here](https://www.kaggle.com/code/nicapotato/keras-timeseries-multi-step-multi-output/notebook))
* The effect is the scale changes for prediction
* Original 
* Prediction (Notice the scale) 
* For rainfall prediction (Notice the scale on either sides left and right)
* Notice that the scales changes drastically
* I tried to manually correct the scale using standardization of two scales i.e on original dataset and the prediction
* #Scaling up of data
* for x in df.columns:
* if(x == 'DATE'):
* continue
* smin = df[x].min()
* smax = df[x].max()
* cmin = PREDICTION\_TRAIN[x].min()
* cmax = PREDICTION\_TRAIN[x].max()
* PREDICTION\_TRAIN[x] = PREDICTION\_TRAIN[x].apply(lambda x: smin + ((abs(x - cmin)/cmax)\* (smax - smin + 1)))
* Now to predict correct ranged values, it is better to switch to one output LSTM model which is a general way to proceed forward.

Now doing independent training on West bengal dataset:

1. Rainfall
2. Air Temperature
3. Wind 
4. Pressure
5. Humidity