Time series Analysis

Aishee dasgupta
Arkaprava Sinha
Rohan Khaitan
Subhasish Basak

CHENNAI MATHEMATICAL INSTITUTE

April 16, 2019

Overview

1. Introduction
2. Statistical keywords & methodologies used
   - Time series data:
   - Stationary & Non-stationary component of a Time series:
3. Augmented Dicky-Fuller (ADF) test for stationarity:
4. Partial autocorrelation function
5. Auto regressive process of order p(AR(p)):
6. Moving Average process of order q(MA(q)):
7. Autoregressivemoving-average model(ARMA):
8. AutoregressiveIntegrated-moving-average model(ARIMA):
9. Implementation
   - Data Description
   - Hourly Data Visualisation
   - Stationary Test
   - ACF-PACF plot for the daily data
   - ARIMA Model fit
   - Problem in fitting ARIMA model

Introduction

High frequency data refers to time-series data collected at an extremely fine scale.

Generic Time series models do not work well with high frequency data.

Time series data:

- **1.Time series data:**
  Data observed collected or recorded over time is called Time series data. For this project the collected dataset on energy consumption is an example of time series data. The simplest mode of diagrammatic representation of Time series data is the use of Time series plot. Taking two perpendicular axes of co-ordinates, the vertical one for the variable under study) & the other for time points, each pair of values is plotted. The resulting set of points constitutes the Time series plot.

Stationary & Non-stationary component of a Time series:

- **2. Stationary & Non-stationary component of a Time series:** Any time series is generally composed of a Non-stationary part N_t and a Stationary part S_t. If $S_t = 0$ the time series is said to be purely Non-stationary. If $N_t = 0$ then it is called a purely stationary series. Most time series observed in practice are Non-stationary & Stationary both superimposed.

Statistical keywords & methodology used

- **Non-stationary component of a Time series:**
  Non-stationary as the name suggests, the changes in the values of a time series variable over a period of time is due to the Non-stationary component present in that time series. Analysis of Non-stationary in time series calls for:
  - Identification of the components of which the time series is made up of.
  - Isolation and measurement of the effects of these components separately and independently holding the other effects constant.

Statistical keywords & methodology used

- **Stationary component of a Time series:**
  - **strongly stationary**
  - **weakly stationary**

Stationary component of a Time series

- **Strongly Stationary**

  A time series Y_t is said to be strongly stationary if

  $$\{Y_{t1}, Y_{t2}, \cdots, Y_{tk}\} \xrightarrow{D} \{Y_{t1+h}, Y_{t2+h}, \cdots, Y_{tk+h}\}$$

  i.e. $\{Y_{t1}, Y_{t2}, \cdots, Y_{tk}\}$ converges to $\{Y_{t1+h}, Y_{t2+h}, \cdots, Y_{tk+h}\}$ in distribution for any time points $t1, t2, \cdots, tk$ and $\forall$ h

Stationary component of a Time series

- **Weakly Stationary**

  A time series Yt is said to be weakly stationary if its statistical properties do not change over time i.e.

  - $E(Y_t) = E(Y_{t+h}) = \mu(say) \forall h$
  - $V(Y_t) = \sigma^2(say)$ is constant
  - $Cov(Y_t, Y_{t+h}) = \gamma_h(say)$, i.e covariance depends only on lag.

Method of Classical decomposition:

To analyse the Non-stationary part of the time series in hand, this method is applied. In the Classical approach of Time series analysis it is assumed that any time series Y_t is composed of four major components viz.

- Trend (T_t)
- Seasonal component (S_t)
- Cyclical component (C_t)
- Irregular component (I_t)

$$Y_t = f(T_t, S_t, C_t, I_t)$$

The idea is to identify the functional relationship between these components by visual inspection graphical analysis of the data and estimate each of these components separately using proper statistical tools. Then using the functional relationship, the estimated components are combined together to provide a estimate of the original time series.

- **Trend:**
  rend is the smooth, regular, long term movement of the time series observed over time. It may be an upward, downward movement or remain at a constant level, but sudden of frequent changes are incompatible with the idea of trend.
- **Seasonal component:**
  The seasonal component is the periodic movement in a time series which recurs or repeats at regular intervals of time and where the period is not longer than a year.
- **Cyclical component:**
  The Cyclical component is the oscillatory movement in time series, the period of oscillation being more than one. The length of a cycle and also the intensity of fluctuation may vary from one cycle to another.
- **Irregular component:**
  It is the component which is either wholly unaccountable or are caused by such unforseen events as was, floods, strikes etc.

Augmented Dicky-Fuller (ADF) test for stationarity:

The Augmented DickeyFuller test(ADF), tests the null hypothesis that a unit root is present in a time series sample. A unit root is a feature of some stochastic processes such that, a linear stochastic process has a unit root if 1 is a root of the process's characteristic equation. Such a process is non-stationary but does not always have a trend.

Augmented Dicky-Fuller (ADF) test for stationarity:

Thus it is equivalent to state the Hypothesis of interest of an ADF test as,
H_0: The series is non-stationary.
H_1: The series is stationary.

Augmented Dicky-Fuller (ADF) test for stationarity:

For a given time series y_t ,the ADF test considers the model,

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} \cdots + \delta_{p-1} \Delta Y_{t-p+1} + \epsilon_i$$

Where,α is a constant, β is the coefficient on a time trend and p is the lag order of the autoregressive process.

Imposing the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk and using the constraint $\beta = 0$ corresponds to modelling a random walk with a drift.

ADF test for stationarity

The unit root test is then equivalent to test,
$H_0 : \gamma = 0$
$H_1 : \gamma < 0$
The appropriate test statistic is given by, $D = \frac{\hat{y}}{S.E(\hat{y})}$
Test rule: Reject H_0 iff $D_{obs} < D_{crit}$ (obtained from DickeyFuller table) .

he intuition behind the test is that if the series is integrated then the lagged level of the series y_{t-1} provide no relevant information in predicting the change in y_t besides the one obtained in the lagged changes y_{t-k} Δy_{t-k}. In this case the null hypothesis is not rejected.

Autocorrelation plot (ACF):

For a given time series Y_t the Autocorrelation function is given by

$$\rho_h = \frac{Cov(Y_t, Y_{t+h})}{V(Y_t)}$$

The graph of the autocorrelation function against ρ_h gives the Autocorrelation plot. The graph expresses how correlation between any two values of the time series changes as the extent of separation in time changes. Observing the distinctive shape of the sample ACF one can identify the underline probability model.

Now for a given time series data $y_1, y_2, \cdots, y_T$, the sample ACF is the estimate given by,

$$r_h = \hat{\rho}_h = \frac{C_h}{C_0}$$

where, $C_h = \sum_{t=1}^{T-h} \frac{(y_t - \overline{y})(y_{t+h} - \overline{y})^2}{T}$ $C_0 = \sum_{t=1}^{T} \frac{(y_t - \overline{y})^2}{T}$

Partial autocorrelation function

In time series analysis, the partial autocorrelation function (PACF) gives the partial correlation of a stationary time series with its own lagged values, regressed the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control for other lags.

Partial autocorrelation function

Given a time series z_t partial autocorrelation of lag k, denoted $\alpha(k)$ is the autocorrelation between z_t and z_{t+k+1} that is not accounted for by lags 1 to k , inclusive.

$$\alpha(1) = corr(z_2, z_1), \text{for } k = 1$$

$$\alpha(k) = corr(z_{t+k+1} - P_{t,k}(z_{t+1})), \text{for } k \geqslant 2$$

where $P_{t,k}(x)$ surjective operator of orthogonal projection of x onto the linear subspace of Hilbert space spanned by $x_{t+1}, \cdots, x_{t+k}$

Auto regressive process of order p(AR(p))

Let $\{X_t\}$ be a purely random process i.e. X_ts are identically independently distributed with mean 0 and variance σ^2, then $\{Y_t\}$ is said to be an AR(p) process if,

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \cdots + \alpha_p Y_{t-p}$$

Here $\alpha_1, \alpha_2, \cdots, \alpha_p$ are the p unknown parameters of the process.

Moving Average process of order q(MA(q))

The moving-average model specifies that the output variable depends linearly on the current and various past values of a stochastic term.
Let $\{X_t\}$ be a purely random process i.e. X_ts are identically independently distributed with mean 0 and variance σ^2, then $\{Y_t\}$ is said to be an MA(q) process if,

$$Y_t = \mu + \theta_1 X_{t-1} + \theta_2 X_{t-2} + \cdots + \theta_q X_{t-q}$$

Here $\theta_1, \theta_2, \cdots, \theta_q$ are the q unknown parameters of the process.

Autoregressivemoving-average model(ARMA)

The notation ARMA(p, q) refers to the model with p autoregressive terms and q moving-average terms. This model contains the AR(p) and MA(q) models,

Let $\{X_t\}$ be a purely random process i.e. X_ts are identically independently distributed with mean 0 and variance σ^2, then $\{Y_t\}$ is said to be an ARMA(p,q) process if,

$$Y_t = c + X_t + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{i=1}^{q} \theta_i X_{t-i}$$

AutoregressiveIntegrated-moving-average model(ARIMA)

In statistics and econometrics, and in particular in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model.

- AR - Auto Regressive
- I - Integrated/Differencing
- MA - Moving Average

Data Description:

- Energy Consumption Data - East Kentucky Power Cooperative (EKPC) - estimated energy consumption in Megawatts (MW)
- June 2013 to August 2018
- About 46K data point
- Hourly Data

Data Visualisation:

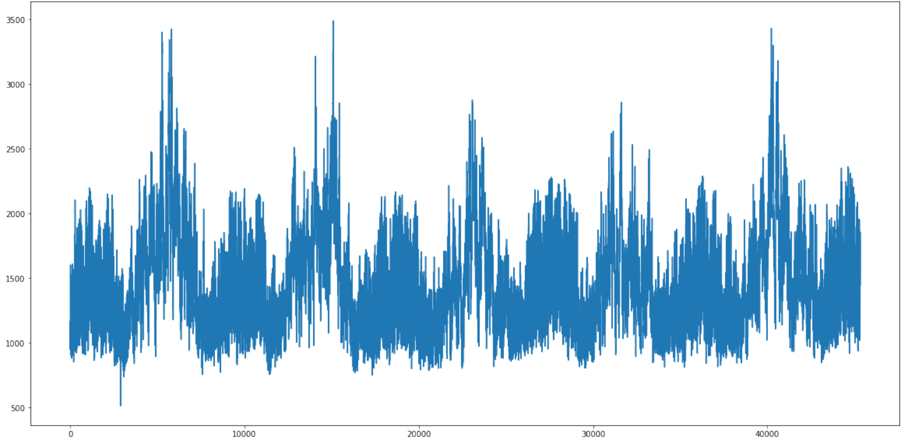


Figure: Hourly Energy Consumption

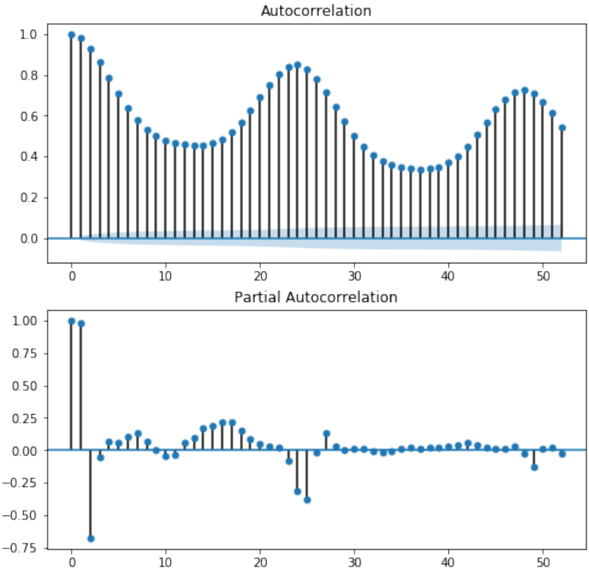
Stationary Test:

- Augmented Dicky-Fuller Test
- P-Value - 2.895578e-19
- Null Hypothesis Rejected
- Stationary Data

Python Models:

- Pyramid ARIMA - Auto ARIMA
- Statsmodels - ARIMA
- SARIMA

ACF-PACF plot for the raw data :



ARIMA Model fit :

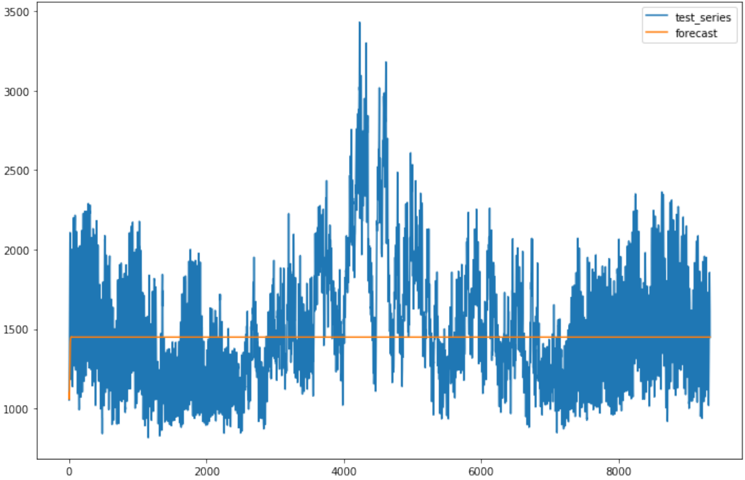


Figure: Prediction on Hourly Test Data

Hourly to Daily Data!

- 24 timepoints in a day
- Sum over all values for the timepoints
- Data Size reduces to 1900

Data Visualisation:

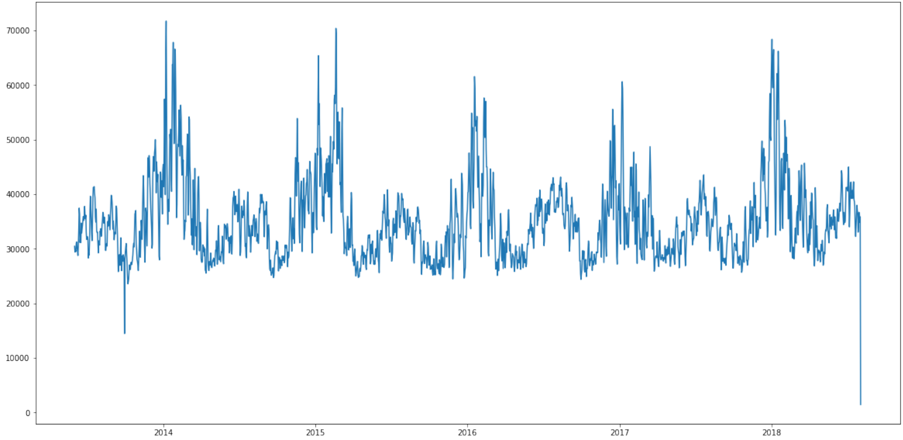
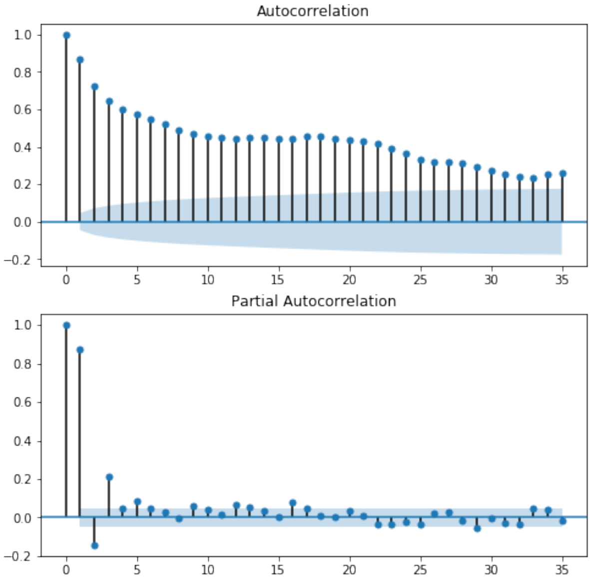


Figure: Daily Energy Consumption

Stationary Test:

- Augmented Dicky-Fuller Test
- P-Value - 0.000171
- Null Hypothesis Rejected
- Stationary Data

ACF-PACF plot for the daily data :



ARIMA Model fit :

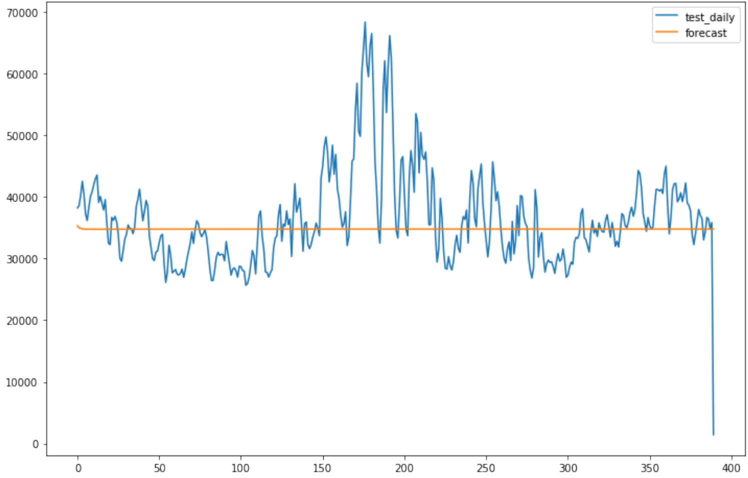


Figure: Prediction on Daily Test Data

Problem in fitting ARIMA model:

- Can not capture all the seasonality together
- Monthly Data - No of days in a month!
- Yearly Data - Leap Year!
- Holidays!

Prophet:

Prophet is an open source library published by Facebook that is based on decomposable (trend+seasonality+holidays) models. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters and has support for including impact of custom seasonality and holidays!

Why Prophet?

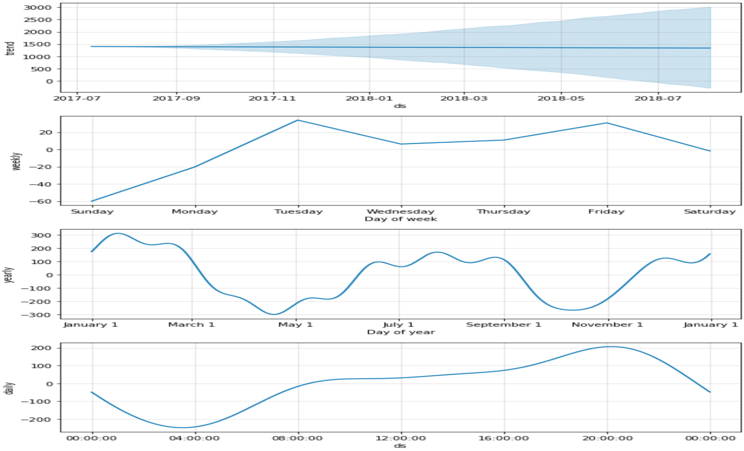
- Hourly, daily, or weekly observations with at least a few months of history
- Important holidays that occur at irregular intervals that are known in advance
- A reasonable number of missing observations or large outliers
- Historical trend changes, for instance due to product launches
- Trends that are non-linear growth curves, where a trend hits a natural limit or saturates

How Prophet Works?

At its core, the Prophet procedure is an additive regression model with four main components:

- A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting changepoints from the data.
- A yearly seasonal component modeled using Fourier series.
- A weekly seasonal component using dummy variables.
- A user-provided list of important holidays.

Seasonality Captured!



Prophet Fit:

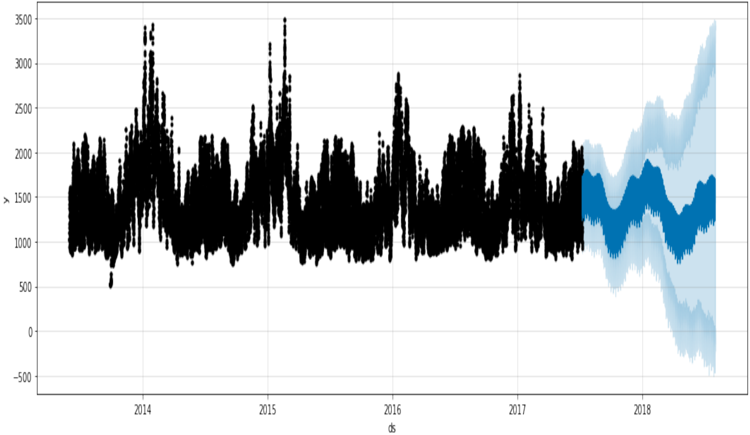


Figure: Train and Predicted Test Data

Prophet Fit:

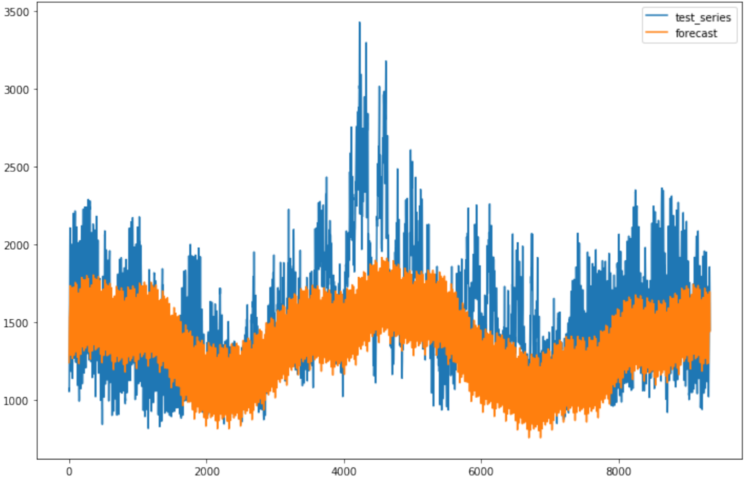


Figure: Test Data and Predicted Values

Can we do better?

- ARIMA fits very badly
- Prophet works relatively better

LSTM - What it is

- Long Short Term Memory networks usually just called LSTMs are a special kind of RNN, capable of learning long-term dependencies.
- LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

LSTM - Architecture

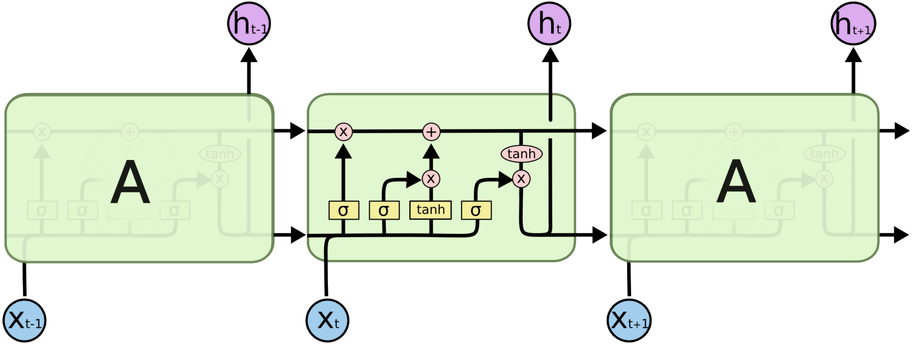
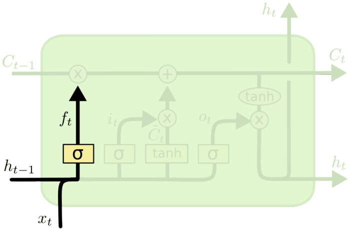


Figure: LSTM

LSTM - Architecture

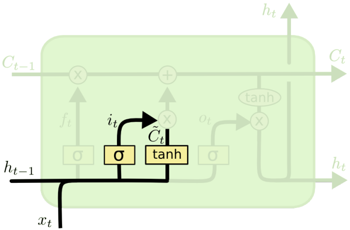


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

The first step in our LSTM is to decide what information were going to throw away from the cell state. This decision is made by a sigmoid layer called the forget gate layer. It looks at h_{t-1} and x_t and outputs a number between 0 and 1 for each number in the cell state C_t.

LSTM - Architecture

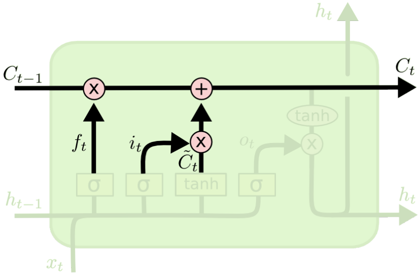
The next step is to decide what new information were going to store in the cell state. This has two parts. First, a sigmoid layer called the input gate layer decides which values well update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$ that could be added to the state. In the next step, well combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

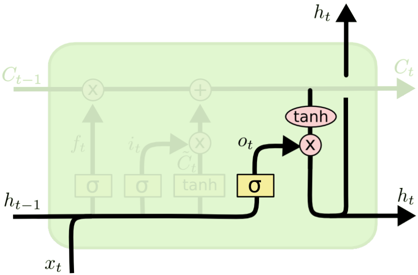
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$ .This is the new candidate values, scaled by how much we decided to update each state value.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Finally, we need to decide what were going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state were going to output. Then, we put the cell state through tanh (to push the values to be between 1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



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

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

LSTM Fit:

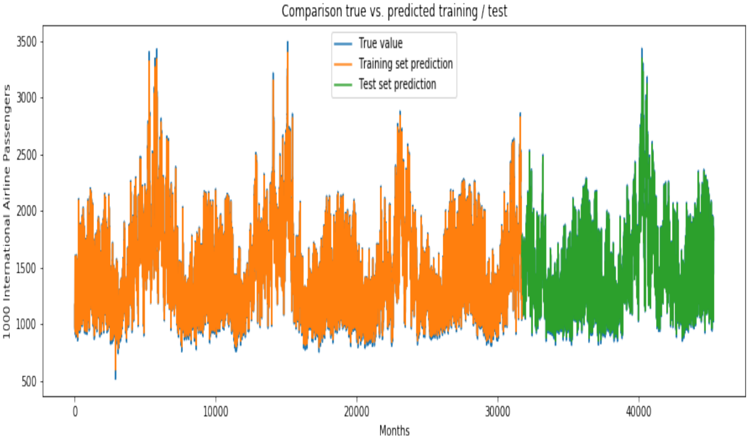


Figure: Actual and Predicted Values

Summary

- Generic time series models fail to work on high frequency data
- LSTM seems to give the best fit to the data
- With the forecast estimates can be made about the actions taken like incorporating renewable sources of energy, etc.
- This model can be used for other high frequency time series data like temperature etc.