SAPIENZA
UNIVERSITÀ DI ROMA

Short-term water demand forecasting using traditional and deep learning models

Facoltà di Ingegneria dell'informazione,informatica e statistica

Corso di Laurea Magistrale in Data Science

Candidate

Poompavai Chandraprakash

ID number 1853976

Thesis Advisor

Prof. Simone Scardapene

Co-Advisor

Dr. Pietro Dicosta

Academic Year 2021/2022

Thesis defended on January 2022
in front of a Board of Examiners composed by:

Data Science Committee Members (chairman)

Short-term water demand forecasting using traditional and deep learning models

Master's thesis. Sapienza – University of Rome

© 2022 Poompavai Chandraprakash. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: cmpavaichandru@gmail.com

Abstract

In today's world, we could see enormous amounts of big data in every domain. Time series forecasting is one such domain, which uses enormous amounts of sensor data with the application of artificial intelligence, and data science. Analysing time series data with effective visualization can help to produce various insightful inferences.

The project involves short term water demand forecasting using traditional and deep learning models. This prediction helps to identify and plan for seasonality, annual patterns, production capacity, and expansion over a longer period. which in turn drives a long-term business strategy (e.g., plans to launch a facility or store internationally and expand into new markets).

External factors are also considered like weather conditions, to check the water demand forecasting in a shorter term. Both univariate and multivariate time series analyses are carried out using Long Long short-term memory (LSTM), Seasonal Auto-Regressive Integrated Moving Average with exogenous factors(Sarimax), Face book prophet(Fbprophet), Baseline persistence model.

Acknowledgements

I would like to take this time to offer my gratitude to my thesis supervisor Prof. Simone Scardapane for having allocated time on his busy schedule to supervise this thesis. The door to Prof. Simone's office was always open whenever I ran into trouble or had a question about my research. He consistently welcomed me and steered me in the right direction during my work and writing of this thesis. I further offer many thanks to my co-advisor from NTT Data Italia Dr. Pietro Dicosta without which the thesis was not going to be possible. Without his passionate participation and input, validation and advice, the implementation would not have been a success.

Contents

Chapter 1

Introduction

This Chapter describes the main goals of this dissertation. Adding to that we provide the objectives and motivations for carrying out our study. A time series is a sequence of data points that occurs over some period of time in successive order. In general, time series allows us to find out the influence relationship of factors with the variables over time to time.

1.1 Motivation

With the use of decision-making strategy and managerial skills with respect to time and money over the business sector, we could reach our goals in terms of the fact, "Time is money". The forecast is needed in making strategic decisions under uncertainty. We may not think that we are forecasting, but our choices will be directed by our anticipation of the results of our actions or inactions. Failures are caused by delays and improper/no decisions.

Today, Artificial Intelligence (AI) and Big Data have redefined business forecasting methods. The most common analysis encountered is the time series analysis. Most organisations use time series forecasting for their business strategies. These methods have been used to monitor, clarify, and predict certain ‘cause and effect’ behaviours. Basically, time series analysis helps to understand how the past influences the future.

1.2 Objectives and Contributions

The main objective of the thesis is to perform a time series forecast, for planning the business in terms of seasonality, annual patterns and so on.

The project involves short term water demand forecasting using traditional and deep learning models. Imagine, a hydraulic network with different DMA's (distribution management areas/dma/DMA). In short, these DMA's denote the small clusters of water users (both domestic and commercial property our ages), with a provision to individually monitor the water supplied and consumed. Our goal was to predict the water demand for each DMA, present in a single hydraulic network. Through this prediction, we could analyse the behaviour and the working property of a single DMA, which in turn leads to the implementation of business models for the remaining DMA's found in the entire hydraulic network. This prediction helps to identify and plan for seasonality, annual patterns, production capacity, and expansion over a longer period. This in turn drives a long-term business strategy (e.g., budget and service plans of the project, plans to launch a facility or store internationally and expand into new markets). External factors are also considered like weather conditions, to check the water demand forecasting in a shorter term. We used LSTM, Sarimax, Fbprophet, Baseline persistence model for both univariate and multivariate analysis.

Here we performed hourly, weekly, daily predictions using models such as LSTM, Sarimax, FB prophet and compared them with the baseline resistance model.

1.3 Thesis Outline

The dissertation is divided into different chapters. Each chapter presents the contents with a focus on the title of the chapter. Chapter 2 deals with the literature review of past research and the concepts to solve the problem. Chapter 3 deals with implementation, this section deals with the methodologies to approach the problem, data collection and analysis of data set, experimental models performed to solve the problem. Chapter 4 deals with the discussion of the results. In chapter 5, we provide conclusions of our research as regards the experimental results and we also provide details about our future research directions indicating the methods we intend to employ for future research.

Chapter 2

Literature Review

This chapter describes the models used in univariate and multivariate time series forecasting. However, we put much focus on the comparative study of different models (both traditional and deep learning) in various aspects of time intervals, say hourly, weekly and monthly predictions.

2.1 Baseline model

A baseline model is essential in time series forecasting. It helps to get the knowledge on the performance of the models on the given problem. It provides the point of comparison, models which perform less than baseline models are omitted However there are a few techniques to be considered for creating a proper baseline model as below:

- **Simple:** A method that requires little or no training or intelligence.
- **Fast:** A method that is fast to implement and computationally trivial to make a prediction.
- **Repeatable:** A method that is deterministic, meaning that it produces an expected output given the same input.

2.2 Persistence model

It is the simplest and baseline method to predict time series forecasts. Suppose consider two-time frames such as present t and future t + Th. The baseline is used to calculate the future on the assumption, with respect to the unchanged condition between the present time t and future time t + Th. For stationary time series, when mean and variance do not change over time, which is called dull persistence.

The Persistence model uses the previous day (or the corresponding day in the previous week) as a prediction. Such a method is sensitive to a method that is sensitive to rare events (e.g., the day in the last month is the Black Friday day).

2.2.1 Persistence Naive forecast

The persistence algorithm uses the previous time step(t-1), to forecast/predict the outcome at the next time step(t+1).

Steps to achieve naive forecast:

1. Convert the univariate/multivariate data set into a supervised learning problem.
2. Split the data set into train and test sets.
3. Create a persistence model.
4. Do forecast/prediction and form a baseline performance to create a comparison with other models.

2.3 Time Series concepts

This section deals with general time series concepts such as stationarity, stochastic process, linearity in time series models, deterministic time series.

2.3.1 Stationarity in time series

Analysing time series data involves checking its stationarity. Stationarity is an important concept in time series analysis. It means that the statistical properties of a time series do not change over time. Additionally, stationarity is important because many useful analytical tools and statistical tests and models rely on it. The stationarity in the data set is checked using the Augmented Dickey-Fuller test and Kwiatkowski-Phillips-Schmidt-Shin test.

2.3.2 Stochastic in time series

A Stochastic process is a statistical phenomenon, it has the collection of random variables ordered in time. To analyse time series, a stochastic process is considered as a model. This process is considered to generate an infinite collection of all possible observed time series. Every member of the collection is the realisation of the stochastic process.

2.3.3 Deterministic in time series

A time series is one that can be expressed explicitly by an analytic expression. It has no random or probabilistic aspects. In deterministic time series, past and future values are specific to values of a given time.

2.3.4 Linearity in time series

A model is linear if the current value of the series is a linear function of past observations. In the literature, Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) are two linear time series models as described in section 2.4. A model is non-linear, reflecting a nonlinear function of the past observations. To predict volatility changes in time series, non-linear models are used.

2.4 LINEAR Time series models

To design the covariance structure in time series, linear time series models are used. This leads to autoregressive moving average models, formed by two sub-models such as (a) the autoregressive and (a) the moving average models.

2.4.1 Autoregressive Model

This model uses the past behaviour of variable y(t), to predict its future behaviour. The prediction of y(t+1) is considered by adding the weighted values, that y(t) took place in the previous time period along with the error term $\epsilon(t)$.

Autoregression model

In general, the observation at various time slots, such as t-1,t-2....t-n, impacts the time period t. In such a case, the coefficient

factors are used at a particular period of time to find the impact of previous time slots.
For example, the new iPhone model 'I' created by apple depends on all the previous models in the time series. So in general, Auto Regression (AR) model is defined in such a way that the model calculates the regression of past time series and calculates the present or future values in the series.

For example, we can predict the value for the next time step (t+1) given the observations at the last two time steps (t-1 and t-2). As a regression model, this would look as follows:

$$X(t+1) = b0 + b1 * X(t-1) + b2 * X(t-2)$$

Additionally, the model is noted with its order, p as the parameter. Here is an example: AR(p), when p=1 $\rightarrow$ AF(1), is called the first-order autoregression model. This AR method is suitable for univariate time series without trend and seasonal components.

Moving average model

Here external factors play an important role in impacting time period t under different various slots t-1,t-2,t-3... In short, these external factors are termed errors or residuals. The previous time spots impacts are calculated by the coefficient factors at the particular time slot.

For example, the stock market of the company 'I' depends on any merger with other companies that happened overnight or the company's demolition due to money factors such as bankruptcy and so on.

So, henceforth in a time series, this moving average model calculates the external factors namely errors or residuals of the past and in turn, calculates the present or future values in the series as shown below.

$$X_t = \alpha_1 * \epsilon_{t-1} + \alpha_2 * \epsilon_{t-2} + ...\alpha_k * \epsilon_{t-k}$$

Additionally, the model is noted with its order, q as the parameter. Here is an example: MA(q), when q=1 $\rightarrow$ MA(1), is called the first-order moving average model. This method is suitable for univariate time series without trend and seasonal components.

Autoregressive Moving Average

It is also called the ARMA model, which is the combination of both autoregression(AR) and moving average(MA) models. So here, to forecast the future values, the impact of previous lags along with errors or residuals are considered here.

So henceforth let α and β be considered as the coefficients of the MA and AR model. The representation would be as follows:

$$X_t = \beta_1 * \times_{t-1} + \alpha_1 * \epsilon_{t-1} + \beta_2 * \times_{t-2} + \\ \alpha_2 * \epsilon_{t-2} + ...\beta_k * \times_{t-k} + \alpha_k * \epsilon_{t-k}$$

Additionally, the model is noted with its order, p as the parameter for autoregressive model AR(p) and q as the parameter for moving average model MA(q) respectively. For example, ARMA(p,q), when p=1, q=1 $\rightarrow$ ARMA(1,1), is called the first-order autoregressive moving average model. This model is suitable for univariate time series without trend and seasonal components.

2.4.2 Autoregressive Integrated Moving Average

The first and foremost thing to be considered in time series is stationary. So, we need to convert the series to stationary time series. Here comes the picture of an integrated method usage, which can be achieved by subtracting the t-n value from the time series t, where n = 1,2,3...This is also called a differencing method and based on the n value, it is called first-order differencing with n=1. When n=2, it is called second-order differencing and so on. In the end, the main goal is to achieve stationary time series.

So, the ARIMA model is similar to the ARMA model, except for the fact it includes the integrated or differencing term 'I'. ARIMA model combines both AR and MA models as well as an integrated/differencing preliminary processing step of the sequence to make the time sequence stationary, this process or parameter is called Integration(I). Here it also considers the previous lags along with residual errors for future forecasting.

Additionally, the model is noted with its order, p as the parameter for autoregressive model AR(p) and q as the parameter for moving average model MA(q) and I as the parameter for integration or differencing (I) respectively. For example, ARMA(p, q, I), when p=1, q=1, I=1 $\rightarrow$ ARMA(1,1,1), is called the first-order autoregressive moving average differencing model. This model is suitable for univariate time series with trends and without seasonal components.

Example : ARIMA(0, 1, 0) model (or I(1) model) is given by $X_t = X_{t-1} + \varepsilon_t X_t = X_{t-1} + \varepsilon_t - whichissimplyarandomwalk.$

2.4.3 Seasonal Autoregressive Integrated Moving-Average

Sarimax stands for Seasonal Autoregressive Integrated Moving average. It is an updated version of the ARIMA model. SARIMAX includes seasonal effects and exogenous factors along with ARIMA components in the model. So basically, with prior time steps, the SARIMAX model produced the output as the linear function of the difference in observations and seasonal observations, errors and seasonal errors.

This model holds a seasonal pattern, with its relation with external factors. This speciality in the model differs from other models. For instance, for climatic time series, the temperature holds seasonal relationships such as low in winter or high in summers. Additionally, this also includes external factors like humidity, the temperature in winter is increased due to rain, with the chance of low temperature. So the prediction of exact value with these factors would not be possible if there is no cyclical or seasonal behaviour. So comparatively other models can deal with this type of data.

The sarima configuration is simple with three trend elements as such:

1. Trend autoregression order(p)
2. Trend difference order (d)
3. Trend moving average order (q)

Similarly, concerning seasonal components, there are four elements as such:

1. Seasonal autoregressive order(P)
2. Seasonal difference order(D)
3. Seasonal moving average order(Q)
4. Time steps, considered for the single seasonal period(m)

So in a nutshell, Sarima holds the below syntax:

$$SARIMA(p, d, q)(P, D, Q, m)$$

Example: SARIMA(1,0,1)(1,2,2,6) here, m= 6 depicts, in a year there are 6 months seasonality, and this m value has a huge impact on PDQ too. In other words, seasonal offset observation in a model is seen concerning P values. Additionally, D=2 leads to the calculation of second-order seasonal difference and Q=2, would lead to a calculation of second-order errors in the model.

2.4.4 Box-Jenkins Methodology

The Box-Jenkins method named and developed by two statisticians George Box and Gwilym Jenkins is widely used in time series analysis. It's basically used in ARMA or ARIMA model to find the best fit value of the time series model using the past values, in the time series data. The Box-Jenkins model is a forecasting methodology, which uses regression analysis on time series. In short, the Box-Jenkins is a mathematical model, created to do forecasting of particular range from a time series data. This model can also use different types of time series data for forecasting purposes. One such example of Box-Jenkins is the ARIMA, to predict outcome/ forecast it uses differences between data points. For the trends, it uses AR, MV and seasonal differencing for forecasting. It is created based on the knowledge of the occurrences which have happened in the past to influence future values. There are some drawbacks too with the time limit as this box-Jenkins model is only used for forecasting events with time frames equal to one and half years or less than that. The model approach follows three main steps:

1. **The model identification or selection**: First and primary goal is to make time-series data stationary. by identifying the seasonality in the data set. This can be achieved by checking the ACF and PACF plots respectively. These plots will give a clear picture of which techniques Autoregression (AR) or Moving Average(MA) should be used.

2. **Estimation of parameters**: This deals with computation algorithms such as maximum likelihood estimation or non-linear least square estimation, to reach or attain the coefficients to get the best fit of the model.

3. **Model checking and forecasting**: Our final goal is to get the stationary univariate process. So this step deals with checking the model to reach our final goal. So here come the residuals of the model to the picture, these residuals hold a relationship with time with constant mean and variance. Additionally, these residuals should be independent of each other too. If the parameter estimation is not great, the steps have to be carried out to create a better model. Regarding, Null hypothesis: White noise residuals are seen.

Autocorrelation and partial autocorrelation

ACF and PCF plots are used in time series data for forecasting purposes. These plots help us to find the relationship between the observations of time series at present with the previous lags.

In short, the ACF plot tells the correlation relationship of time series data and its lagged values. This plot is a bar chart that holds the correlation coefficients of the respective. This bar chart is plotted with confidence bands. So basically it tells us how the present value in the time series is correlated with past values 'n' with n = 1,2,3...k.

Adding to that, time series has general components like residual, trend, seasonality, cyclic and this ACF will take into consideration these components while finding the correlations. So all this is put together to form an auto-correlation plot. Dealing with the ACF plot, the x-axis deals with the coefficients of correlation and the y-axis deals with the past lag values. If the two-time series values of time t are y(t-1) and y(t), so to calculate lag 3, it must be the correlation coefficient between y(t) and y(t 3). In statistics, correlation is defined as the relationship between two variables. Each variable takes up the Gaussian distribution, but usually, the relationship between the two variables is calculated using Pearson correlation coefficients, which holds the values from -1(negative correlation) and 1(positive correlation) or 0(no correlation). So with the observations of the present time series, we can calculate the correlation relationship with the observations of previous time series, which are called time steps or lags basically. The autocorrelation function is defined as the plot of the time series autocorrelation by lag.

The partial autocorrelation function often termed PACF, explains the correlation relationship between time series and lags in a partial way. Here residuals are nothing but the leftover, caused by previous lags. In comparison with ACF, these residuals form partial correlation with the future lag values, as the old found variations are already removed before finding the next correlation. This forms a linear regression analysis, in such a way x(t) is calculated from x(t-1),x(t-2) and x(t-3). Additionally, if there is any missed out residual information, that can be modelled by the immediate next lag. So in order to get a good correlation, we can use the feature of immediate next lag, during modelling. One more important issue which arises here is multicollinearity, and it arises when many features are correlated. It is good practice to avoid such multicollinearity. So only the need for features that are relevant, will be considered.

2.5 Facebook prophet

Facebook prophet was developed internally by a Facebook corporation, for time series forecasting works. It is an open-source platform made available to the public in 2017. It was developed by Sean Taylor and Ben Letham in order to overcome a few issues such as inflexibility, extra assumptions and less robustness with respect to existing forecasting methodologies. With much robust tendency, the Facebook prophet is able to produce high-quality forecasts related to business and overcome many forecasting problems as well. So basically Facebook prophet takes a novel approach and predicts forecasting using probabilistic techniques and additive models, and the final outcome is the curve fitting exercise. The Facebook prophet is mainly for business needs with seasonality and general knowledge is required on the events which hold a huge impact on the data set. Such events include Black Friday, Easter holidays, some paid promotional events or the launch of new products in that particular week and so on. In a nutshell, the Facebook prophet uses additive components, to model the time series data.

$$y(t) = g(t) + h(t) + s(t) + \epsilon(t)$$

The above equation is called prophet model components.
Explanation of the model components:

1. y(t): time series prediction.

2. g(t): This is called a trend function. Its aim is to model non-periodic changes using a piecewise linear regression model or linear saturation growth model. This can be tuned using hyperparameter tuning.

3. h(t): This denotes the holidays or special events which impact the data set. This can be tuned using hyperparameter tuning.

4. s(t): This denotes the seasonality in monthly or yearly or weekly or daily or hourly. This factor models the periodic changes in the time series value. Fourier transform plays an important role for modelling and additionally, we can use our own seasonalities using hyperparameter tuning.

5. $\epsilon(t)$: This is the noise component, which affects the model.

This is basically an intuitive approach, which leads to better and conceptual understanding to create predictions. The Prophet is mainly designed for the below business perspectives:

1. It is mainly used for time series data based on additive models.

2. Full-year for historical data, with data being recorded on an hourly, weekly or daily basis.

3. Seasonality plays an important role, and it should be seen on an hourly or weekly or daily basis.

4. Outliers or unavailable data.

5. Irregularities are seen in holiday events that do not necessarily have seasonality.

6. Severe trend changes such as trends within the data set.

Facebook prophet tuning

Despite the fact, Facebook prophet provides high business forecasts with much accuracy, but customisation can be carried out in terms of adjusting or tuning Facebook prophet. Prophet is quite easy with hyperparameter tuning, so additional regressors, seasonalities or special events can be added in hyperparameter tuning.

In a nutshell, prophet follows additive regressive model, which holds several components such as a linear or logistics trend curve, a seasonality curve with daily/annual/weekly basis, holiday or special event curve, re-sampled user-specified curves such as hourly, quarterly and so on. There are four factors that can be considered in Facebook prophet tuning: growth, holidays, changepoints, seasonalities. growth, holidays, changepoints, seasonalities.

Describing the growth, it all says about the plot. If the data plot needs a growing trend with no real saturation insight, then the parameter can be set to 'linear', in other cases, we set the parameter as 'logistics', if the curve of the data plot has saturation. Moving on to holidays, they are time periods where days have the same type of effect for every year. So usually, while performing predictions that have a time frame of festivals in between, this holiday parameter will be considered. Holiday_prior_scale is the other parameter that deals with holidays and this will tell us the effect of holidays on the predictions. So the next parameter is changepoints, these are the points in the dataset when there are sudden or abrupt changes in the trend. There are four hyperparameters for changepoints: changepoints, n_changepoints, changepoint_range and changepoint_prior_scale. Basically, the parameter changepoints will be used, when changepoints dates are supplied, instead of the prophet to have them determined. Once when the own changepoints are provided, the prophet will not estimate any more changepoints. Next comes the seasonality parameter. This parameter can bring great improvements and achieve great insights by changing only a few values during the prophet implementation. This seasonality has a big parameter seasonality_mode, which shows how seasonality components should be integrated with the predictions. It has the default value as additive and multiplicative. It has the default value as the additive and multiplicative. There comes the seasonality_prior_scale parameter. This parameter will again allow your seasonalities to be more flexible.

Stan and its usage in the Facebook prophet

Stan is used in the Facebook prophet library. In general, stan is a platform for statistical modelling and computation with high performance. Usage of stan includes fields such as modelling with statistical data, prediction analysis in various aspects such as business, engineering, biological, social, physical and most widely used in data analysis. Stan's mathematical library provides differentiable probability functions and linear algebra. It also interfaces with popular programming languages in the field of data analysis such as (R, Python, shell, MAT LAB, Julia, Stata) and runs on all major platforms (Linux, Mac, Windows). Stan allows Prophet to create uncertainty levels for future predictions to create the data-driven estimate of forecasting risk and it is possible with Bayesian statistics.

Advantages of the Facebook prophet:

1. Facebook Prophet is used to produce reliable forecast predictions and to meet deadlines and achieve goals and much better performance compared to other models. High-speed forecasting is done with the help of Stan library.

2. Forecast is done without manual intervention and it is done completely in an automation model. The Facebook prophet is more accurate and robust in fields of outliers, missing data or dramatic changes in the time-series data set.

3. With Facebook prophet, the prediction can be tunable in such a way to get the best fit model. Additionally, it requires little domain knowledge in order to achieve the forecast.

4. The library of the Facebook prophet is used in other R and python, at the end, the model is fitted using Stan library.

5. Irregularities are seen in holiday events that do not necessarily have seasonality.

6. Severe trend changes such as trends within the data set.

7. Facebook prophet achieves typical and good results within a short span of time and it's in no way lesser than complicated forecasting techniques with respect to performance. But in order to achieve better performance, hyperparameter tuning is considered.

2.6 Neural networks for time series forecasting

In today's world, neural networks play an important role, as we are dealing with enormous amounts of data. This huge data set is called big data. There are various models including traditional and deep learning models to analyse and handle these data. In the modern world, alongside computer vision, and the internet of things, neural network models play a vital role in various fields such as spam detection, anomaly detection, time series forecasting, future predictions, fraud analysis, medical diagnosis and so on. These deep learning models tend to over-perform traditional models in an effective way in terms of result, speed and performance. These models can be used in both univariate and multivariate time series analysis. Below are a few reasons why deep learning models should be considered in time series forecasting:

1. Deep Learning can deal with huge amounts of data and they are fed into algorithms on further steps.

2. Deep learning uses an end-to-end approach to solve problems.

3. Deep learning performs well in fields such as computer vision, natural language processing, image classification, speech recognition, time series forecasting and so on.

4. In terms of time series forecasting, recurrent neural networks (deep learning framework) is used, which in turn makes reliable predictions on time series data and problems.

5. Deep learning supports multiple inputs and outputs.

6. As we all know, they are structured and unstructured data and Deep learning have the capacity to learn and extract features automatically from the structured and unstructured data set.

7. Deep learning provides more accuracy in terms of prediction and forecasting.

2.6.1 Recurrent Neural Network

Recurrent Neural Network belongs to the class of Artificial Neural Network is often termed as auto-associative or feedback network. RNN is the acronym of Recurrent Neural Network. Here the units are connected to form a directed cycle, leading to showcase the dynamic temporal behaviour. Generally, feed-forward networks deal with data points, which are independent of each other.RNN supports long term dependencies. In such a case, to work on data that has a sequence or holds a dependency comes into the picture of RNN. They involve data sequences where each input is not mapped to output, instead, it creates a function to map input over time to a single output. In a nutshell, RNN has the information of input or previous state in the form of memory in order to create the next output in the sequence. Advantages of RNN of RNN includes the functionality of handling sequence data, the input can be of different lengths, it has the tendency to store the information of the previous state. Disadvantages of RNN include for decision making it does not consider future inputs, vanishing gradient problem and slow computation time. Another important point is the usage of activation functions. It is generally defined as how the weighted sum of the input is transferred into the output from single or multiple nodes in a network layer. There are basically three activation functions used in RNN architecture:

1. TANH: $1/(1+e^{-x})$
2. SIGMOID: $e^x - e^{-x}/e^x + e^{-x}$
3. RELU: max(0,x)

The selection of activation functions has a huge impact on the performance and outcome capability of neural networks and different activation functions can be used in different parts in neural network models.

There are different types of RNN:

1. One to One: Single input x_t produces single output y_t.
2. One to Many: Single input x_t produces multi output such as (y_{t0}, y_{t1}, y_{t2}).
3. Many to One: Many inputs (x_t, x_{t+1}, x_{t+2}) produce a single output y_t.
4. Many to One: Many inputs (x_t, x_{t+1}, x_{t+2}) produce many output (y_{t0}, y_{t1}, y_{t2}).

One such main problem with RNN is the vanishing gradient descent, which clogs the long data sequences. Usually, the information from the past input is carried to the future output and hence the gradients carry this information and there is an update visible in the RNN parameters. So usually, when the gradients become very low, the updating of parameters becomes unusual, at the end leading to the path of non-learning of the network. To overcome this error, LSTM has been implemented.

Types of RNN Architecture:

1. Bidirectional recurrent neural networks (BRNN): Inputs from future time steps are used to improve the network's accuracy.

2. Gated Recurrent Units (GRU): They mainly work on vanishing gradient problems and have reset and update gates. These gates mainly help to consider which data has to be kept for the future.

3. They mainly work on vanishing gradient problems and have three gates namely input, output and forget gates. And these gates decide which information has to be kept for the future.

2.6.2 Long-Short term Memory

Long short term memory is shortly formed as LSTM, which is a kind of recurrent neural network(RNN), which can learn long-term dependencies. They work mainly on vanishing gradient problems and they have three gates namely input, output and forget gates. LSTM can save the backpropagation error, through each and every layer in the network across the time, without losing the information. The cell plays an important role in saving and releasing information as they decide what to store when to store, how much to store and what to release when to release and how much to release. As we know the information is the key in the data, and these gates in LSTM learn by themselves the allowance of this information in three main processes such as entering, leaving and or deletion. All these processes happen in an iterative way of back-propagation error, making decisions, with the help of gradient descent adjusting weights.

In general, the memory cells in LSTM involve both addition and multiplication processes, with respect to the transformation of both input and information. So here comes the flow chain of LSTM working, as the addition process protects the constant error during the backpropagation event. So in order to create the following state with respect to a cell, usually LSTM considers the multiplication process of the current state with new input, but instead, the addition of the current state with new input takes place, while the last cell 'forget' cell still completely depends on the multiplication process.

Chapter 3

Implementation

3.1 Methodology

So basically the ultimate goal of the thesis was to develop a forecasting model, which will help the client to plan and identify seasonality, annual patterns, production capacity, and expansion over the period based on requirements. This in turn leads to the development of business strategy (e.g., plans to launch a facility or store internationally and expand into new markets).

External factors were also considered like weather conditions, football events and so on to check the water demand forecasting in the longer or shorter term. So, in the end, there were two kinds of analysis such as univariate time series analysis and multivariate time series analysis. Our final goal was to do prediction on the data set in three ways as below:

Our final goal was to do prediction on the data set in three ways as below:

1. **Hourly:** This includes hourly prediction on the data set for both the DMA's namely DMA 225 and DMA 468. Data is re-sampled on a daily basis and prediction is carried out for each hour in a day. It is achieved using univariate and multivariate time series analysis.

2. **Weekly:** This includes hourly prediction on the data set for both the DMA's namely DMA 225 and DMA 468. Data is re-sampled on a daily basis and prediction is carried out for each week. It is achieved using univariate time series analysis.

3. **Monthly:** This includes hourly prediction on the data set for both the DMA's namely DMA 225 and DMA 468. Data is re-sampled on a daily basis and prediction is carried out for each month. It is achieved using univariate time series analysis.

So based on the requirement, we worked on two methodologies as below:

1. **Univariate Time Series Analysis:** The term 'univariate' implies that forecasting is based on a sample of time-series observations of the data. They deal with only one variable, along with the timestamp. And we need to do prediction or forecast on this only one variable. The task of this ideology involves playing with the data set and aim to produce visualisation and data plots, comparison of traditional and deep learning models to perform, the forecast prediction on the data set. The analysis was carried out by checking the accuracy metric mean square error. Predictions were done on a weekly, hourly and monthly basis of both DMA's 225 and 468 respectively.

2. **Multivariate Time Series Analysis:** The term 'multivariate' implies that forecasting is based on a sample of time-series observations of the multiple variables in the data set. In general, they contain multiple variables, along with timestamps. And we need to do prediction or forecast on these multiple variables. External factors such as weather conditions or football events can be considered to perform the water demand predictions. With respect to football events, we need to check if the football match has happened on the day and date in that location corresponding to the water demand data set. So, this would follow classification time series analysis. The task of this ideology involves playing with the data set and aim to produce visualisation and data plots, comparison of traditional and deep learning models to perform, the forecast prediction on the data set. The analysis was carried out by checking the accuracy metric mean square error. With respect to external factors, weather conditions are considered here. The prediction was done on an hourly basis.

3.2 Data collection

The below steps describe the data collection part of univariate and multivariate time series analysis.

1. **Univariate Time Series Analysis:** To perform univariate time series analysis Yorkshire data set was considered. This Yorkshire data set is a water data set which records flows of water every 15 minutes to each of the Distribution Management Areas across the region (DMA's). DMA is the acronym of Distribution Management Areas. There are many DMA's found, but we considered two DMA's namely DMA 225 and DMA-468, in the HU1 location of the country. HU1 is a postcode area, located in the Hull postcode-town region, within the county of Yorkshire. The Yorkshire water is mainly used in the HU1 location. Hence to perform the research project we considered distributed management areas found in the HU1 location. This data set contains the flows in l/s for each of the 2000 DMAs that contain domestic properties A breakdown of properties by Postal Sector is given for each of the DMAs in the extract. The dataset followed univariate time series analysis as we found the DMA's recorded the water demand value for the particular date-time. We used one year of data, from April to March mid with regular time intervals of 15 minutes. Here outliers were visualised with respect to the water dataset, but the forecast was done without outlier removal to check the efficiency of the models.

2. **Multivariate Time Series Analysis:** To perform multivariate time series analysis, we combined Yorkshire dataset and Weather dataset. The Weather dataset was taken as the external weather factors, to perform water demand prediction of the DMA at the time interval. WorldWeatherOnline is a company providing accurate and relevant weather data. Their primary weather API types are local weather API, historical weather API, marine, sailing and surfing weather API. Specifically, their historical weather API provides hourly past weather for worldwide locations since July 2008. This dataset followed multivariate time series analysis. We used three months of data, from April mid to July mid with regular time intervals of 15 minutes. Here outliers were visualised with respect to the water dataset, but the forecast

was done without outlier removal to check the efficiency of the models.

3.3 Data Analysis

This section involves the data analysis and observatory part.

3.3.1 Univariate Time Series Analysis Hourly Prediction

This subsection involves the data analysis part of water flows with respect to two DMA's, with the corresponding timestamps.

DMA 225 univariate analysis hourly prediction

Stationarity check

Stationarity check plays an important role in time-series data. There are two hypotheses namely the Null hypothesis (H0) and the Alternate hypothesis (H1). These two are checked in the tests mentioned below. One more factor is the p-value, It is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct.

(a) **Augmented Dickey-Fuller test** The acronym of Augmented Dickey-Fuller is ADF. Null Hypothesis (H0): If failed to be rejected, meaning it is non-stationary. It has some time-dependent structure. Alternate Hypothesis (H1): The null hypothesis is rejected; meaning it is stationary. It does not have a time-dependent structure.

    The analysis is carried on two factors. The first factor is the ADF(Augmented Dickey-Fuller) statistic value. It was -7. The more negative this statistic, the more likely we were to reject the null hypothesis (we have a stationary data-set). Second factor was p-value - $0.000058 < 0.05$. Hence these two values proved data was stationary.

(b) **Kwiatkowski-Phillips-Schmidt-Shin test:** The acronym of Kwiatkowski-Phillips-Schmidt-Shin is KPSS. Null Hypothesis (H0): stationary. Alternate Hypothesis (H1): non-stationary Analysis was carried on KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test value. It was 4.3. The positive value was more statistic, and hence we had a stationary dataset.

Dataset plot

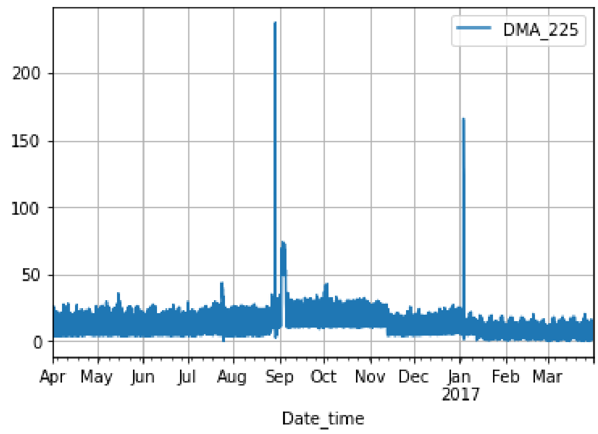

Figure 3.1. DMA-225-univariate analysis

DMA 468 univariate analysis hourly prediction

Stationarity check

Stationarity check plays an important role in time-series data. There are two hypotheses namely the Null hypothesis (H0) and the Alternate hypothesis (H1). These two are checked in the tests mentioned below. One more factor is the p-value, It is the probability of obtaining results at least as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct.

(a) **Augmented Dickey-Fuller test** The acronym of Augmented Dickey-Fuller is ADF. Null Hypothesis (H0): If failed to be rejected, meaning it is non-stationary. It has some time-dependent structure. Alternate Hypothesis (H1): The null hypothesis is rejected; meaning it is stationary. It does not have a time-dependent structure.

The analysis is carried on two factors. The first factor is the ADF statistic value. It was -11. The more negative this statistic, the more likely we were to reject the null hypothesis (we have a stationary data-set). Second factor was p-value - 0.000058 < 0.05. Hence these two values proved data was stationary.

(b) **Kwiatkowski-Phillips-Schmidt-Shin test:** The acronym of Kwiatkowski-Phillips-Schmidt-Shin is KPSS. Null Hypothesis (H0): stationary. Alternate Hypothesis (H1): non-stationary analysis is carried out on KPSS test value. It was 0.2. The positive value was a statistic, and hence we had a stationary dataset.

Dataset plot

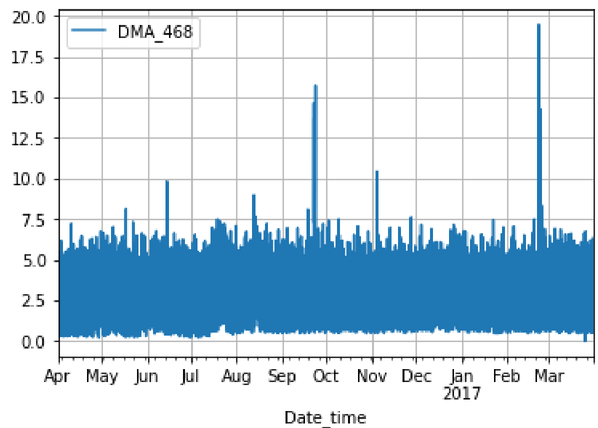

Figure 3.2. DMA-468-univariate analysis

3.3.2 Univariate Time Series Analysis Weekly Prediction

DMA 225 univariate analysis weekly prediction

Time Series decomposition

The below plot depicts the time series decomposition of the DMA 225, on daily re-sampling. This time series decomposition will provide the picture of components such as trend, seasonality and residuals.Data was stationary with ADF static: -2 .81 and KPSS value: 0.63.

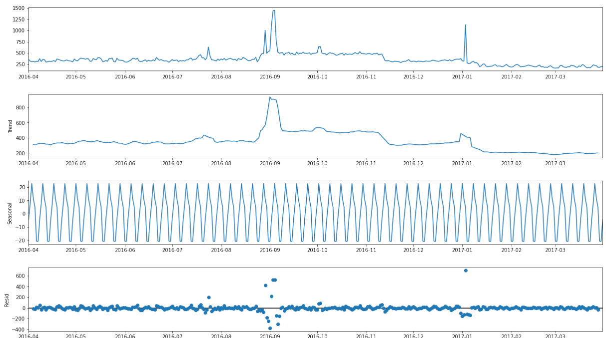

Figure 3.3. DMA-225-Time series decomposition

DMA 468 univariate analysis weekly prediction

Time Series decomposition

The below plot depicts the time series decomposition of the DMA 468, on daily re-sampling. This time series decomposition will provide the picture of components such as trend, seasonality and residuals. Data was stationary with ADF static: -5.8 and KPSS value: 0.14.

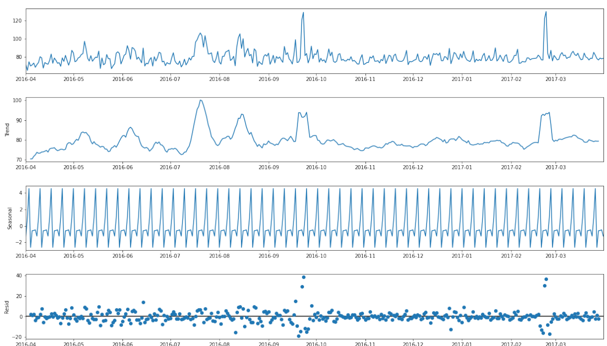

Figure 3.4. DMA-468-Time series decomposition

3.3.3 Univariate Time Series Analysis Monthly Prediction

DMA 225 univariate analysis monthly prediction

Time Series decomposition Here data was resampled on a daily basis, so the final shape of the dataset was (365,1). Data was stationary with ADF static: -2.8 and KPSS value: 0.6.

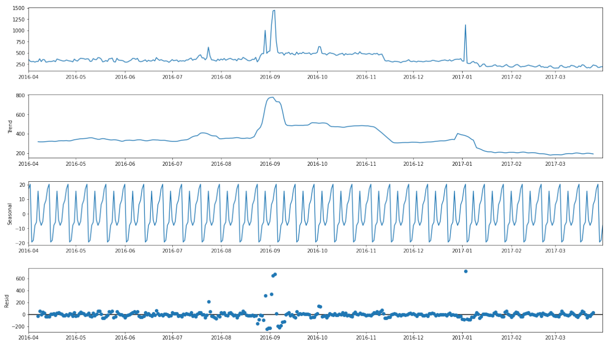

Figure 3.5. DMA-225-Time series decomposition

DMA 468 univariate analysis monthly prediction

Time Series decomposition Here data was resampled on a daily basis, so the final shape of the dataset was (365,1). Data was stationary with ADF static: -5.8 and KPSS static: 0.14

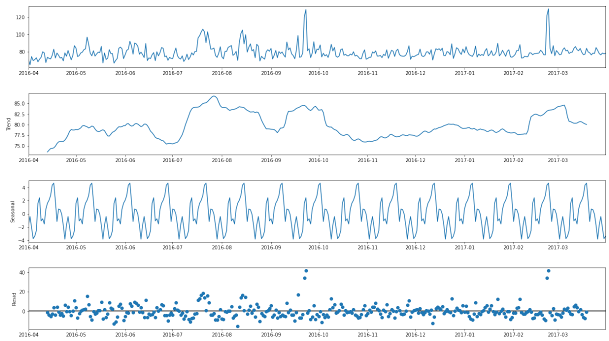

Figure 3.6. DMA-468-Time series decomposition

3.3.4 Multivariate Time Series Analysis hourly prediction

This subsection involves the data analysis part of water flow with respect to two DMA's and weather variables with the corresponding timestamps.

Dataset plot of DMA 225 multivariate analysis and DMA 468 multivariate analysis

The below graph contains the multi variables, both weather and DMA values with respect to timestamp. The weather variables include tempC, HeatIndexC, FeelsLikeC, Wind-ChillC, wind speed mph, sunHour, precipMM, humidity.

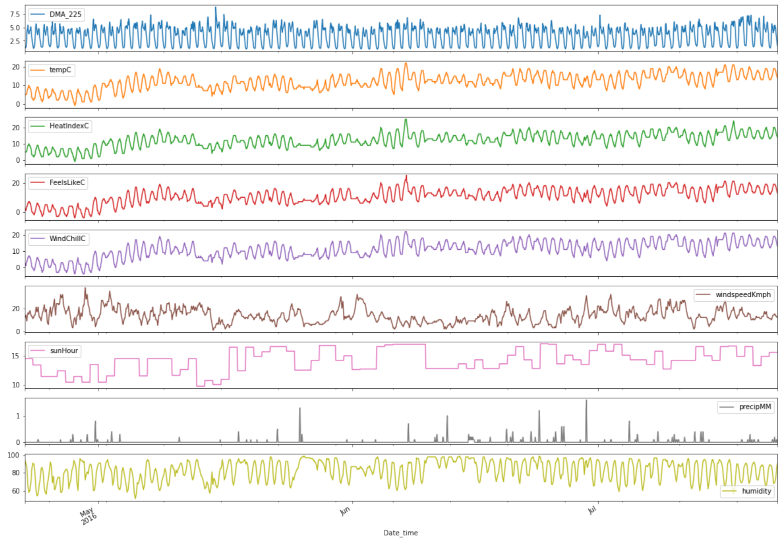

Figure 3.7. DMA-225-multivariate

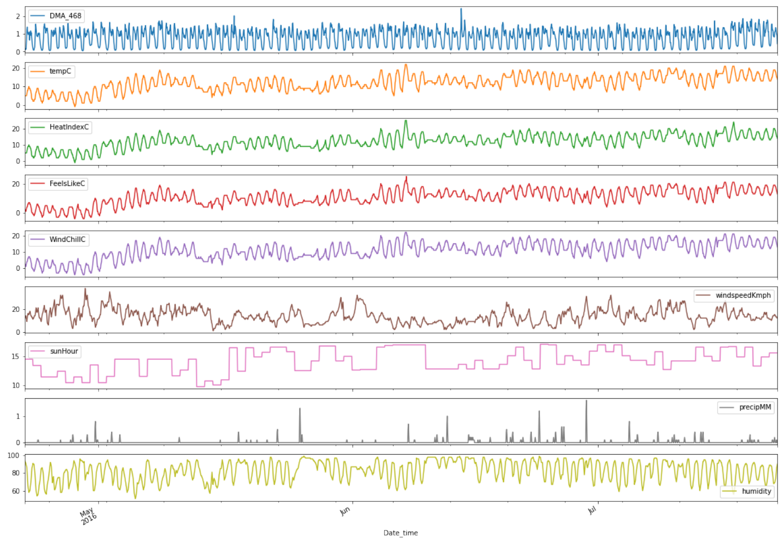

Figure 3.8. DMA-468-multivariate

3.4 Experimental Models

The below section describes the models performed for the carry out univariate and multivariate time series forecast.

(a) Univariate Time Series Analysis - hourly prediction
(b) Univariate Time Series Analysis - weekly prediction
(c) Univariate Time Series Analysis - monthly prediction
(d) Multivariate Time Series Analysis - hourly prediction

3.4.1 Univariate Time Series Analysis - hourly prediction

This includes prediction models implemented on two DMA's namely DMA 225 and DMA 468 respectively on an hourly basis. Data was re-sampled on an hourly basis and the prediction was carried out.

DMA 225 Univariate Analysis: Hourly prediction

To perform univariate analysis on DMA 225, we considered the last 1059 points as test set and the remaining portion of data to train set. We proposed LSTM and Facebook prophet and baseline persistence model on this data proportion and calculated the prediction using root mean square error metrics.

<u>LSTM Model</u>

We used the tensor flow library to perform the LSTM model on the DMA 225 dataset. The ultimate goal of this model was to calculate the hourly predictions. In the LSTM model, the look-back period was considered as 24 hours and output was predicted for 24 hours. So it followed a supervised learning problem approach. We obtained the rmse score of 1.4367 as the result.

<u>Facebook Prophet Model</u>

We used the Facebook prophet library on the DMA-225 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 4.227 as the result.

Baseline Persistence Model

We performed the Baseline persistence model on the DMA-225 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 2.156 as the result.

Plots of LSTM and Facebook prophet - DMA 225 hourly

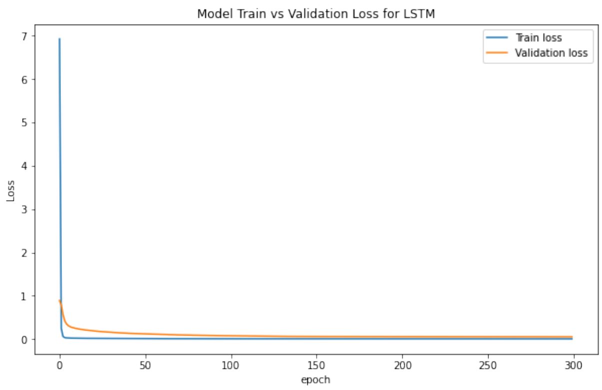

Figure 3.9. LSTM DMA225 univariate hourly prediction: loss plot

Figure 3.10. LSTM DMA225 univariate hourly prediction: original vs predicted plot

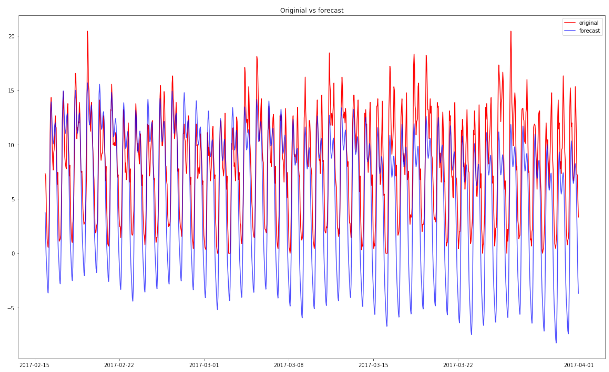

Figure 3.11. Facebook Prophet DMA225 univariate hourly prediction: original vs predicted plot

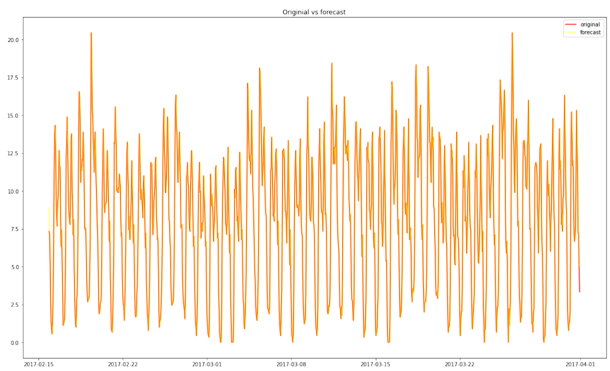

Figure 3.12. Baseline DMA225 univariate hourly prediction: original vs predicted plot

DMA 468 Univariate Analysis: Hourly prediction

To perform univariate analysis on DMA 468, we considered the last 1059 points as the test set and remaining train set. We proposed LSTM and FB prophet and baseline persistence models on this data proportion and calculated the prediction using root mean square error metrics.

LSTM Model

We used the TensorFlow library to perform the LSTM model on the DMA 468 dataset. The ultimate goal of this model was to calculate the hourly predictions. In the LSTM model, the look-back period was considered as 24 hours and output was predicted for 24 hours. So it followed a supervised learning problem approach. We obtained the rmse score of 0.7799 as the result.

Facebook Prophet Model

We used the Facebook prophet library on the DMA-468 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 1.2 as the result.

Baseline persistence Model

We performed the Baseline persistence model on the DMA-468 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 1.084 as the result.

Plots of LSTM and Facebook prophet - DMA 468 hourly

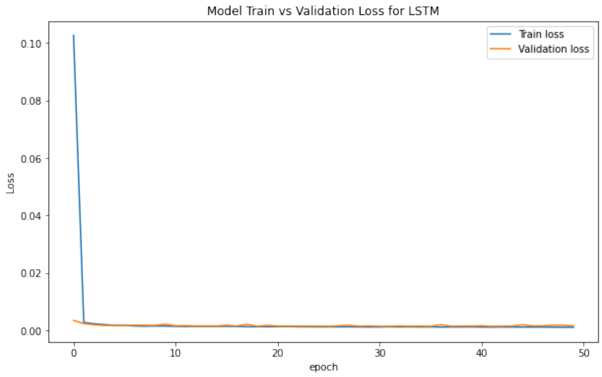

Figure 3.13. LSTM DMA468 univariate hourly prediction: lossgraph

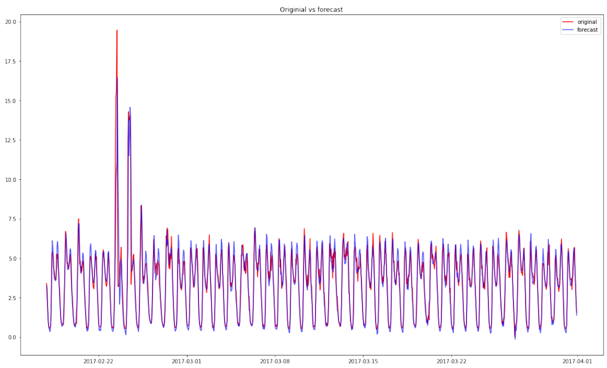

Figure 3.14. LSTM DMA468 univariate hourly prediction: original vs predicted plot

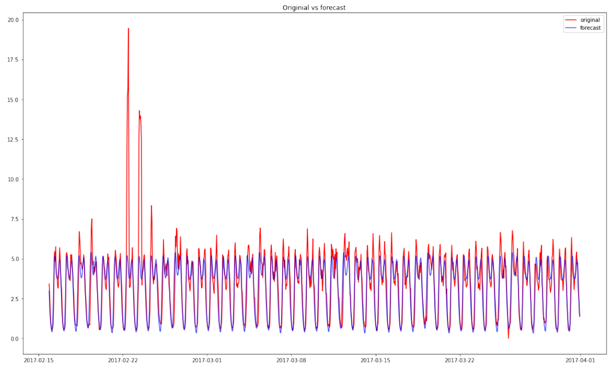

Figure 3.15. Facebook prophet DMA468 univariate hourly prediction: original vs predicted plot

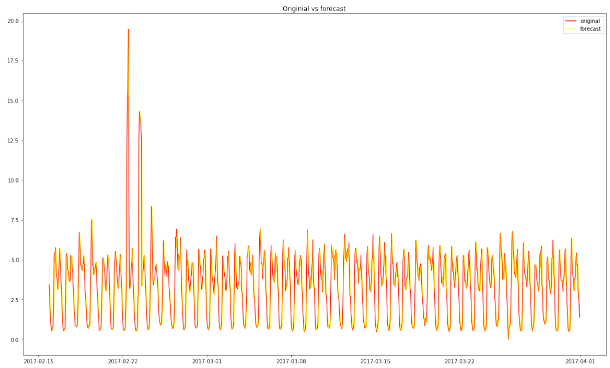

Figure 3.16. Baseline DMA468 univariate hourly prediction: original vs predicted plot

3.4.2 Univariate Time Series Analysis - weekly prediction

This includes prediction models implemented on two DMA's namely DMA 225 and DMA 468 respectively on a weekly basis. Data was re-sampled on a daily basis. We performed models such as LSTM, Sarimax and baseline.

DMA 225 Univariate Analysis: Weekly prediction

To perform univariate analysis on DMA 225 was re-sampled on a daily basis and the length of the dataset was 365. 70% of the dataset was considered as a train set and 30% was considered as a test set. We proposed LSTM and Sarimax and baseline persistence on this data proportion and calculated the prediction using root mean square error metrics.

Baseline Model

Here a simple baseline persistence model was carried out. The rmse value of the baseline model was 113.697.

Sarimax Model

We used sarimax on the DMA-225 dataset. Performed grid search parameters using train set to get best parameters and using those we calculated weekly prediction. Hence finally achieved a root mean square value of 132.542 as the result.

LSTM Model

We used LSTM on the DMA-225 dataset. Here the lookback period of 3 days was considered to perform the weekly prediction. Hence finally achieved a root mean square value of 91.41 as the result.

Plots of Baseline, Sarimax and LSTM - DMA 225 weekly

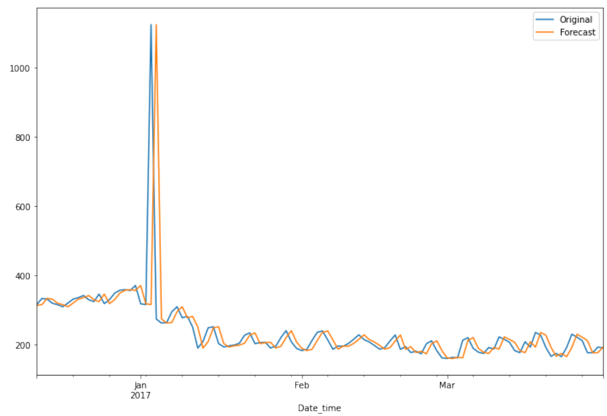

Figure 3.17. Baseline DMA225 univariate weekly prediction: original vs predicted plot

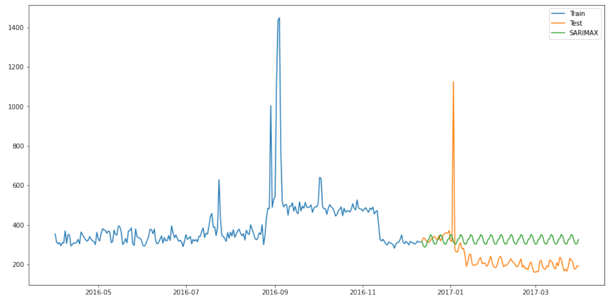

Figure 3.18. Sarimax DMA225 univariate weekly prediction

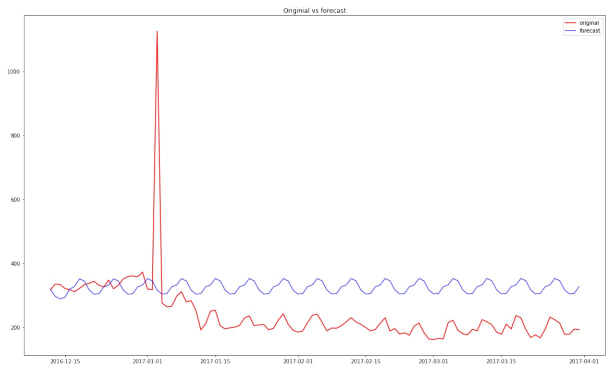

Figure 3.19. Sarimax DMA225 univariate weekly prediction: original vs predicted plot

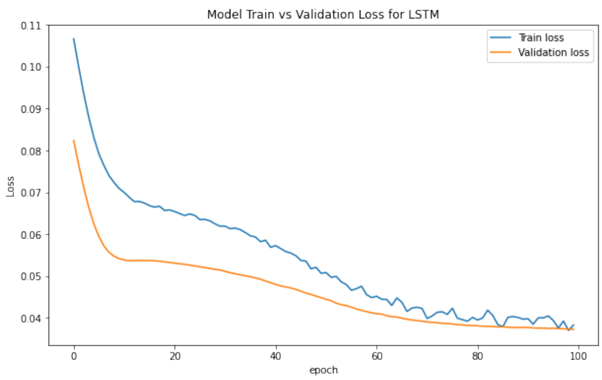

Figure 3.20. LSTM DMA225 univariate weekly prediction: lossgraph

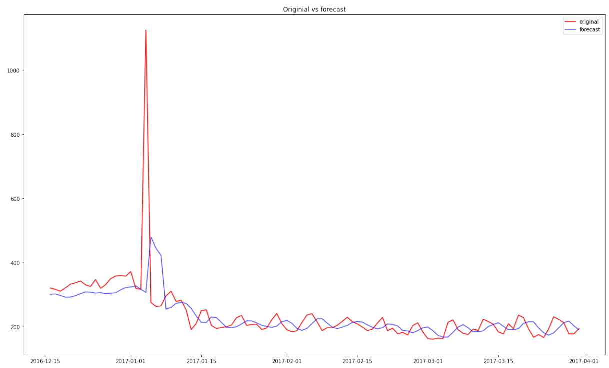

Figure 3.21. LSTM DMA225 univariate weekly prediction: original vs predicted plot

DMA 468 Univariate Analysis: Weekly prediction

To perform univariate analysis on DMA 468 was re-sampled on a daily basis and the length of the dataset was 365. 70% of the dataset was considered as a train set and 30% was considered as the test set. We proposed LSTM and Sarimax and baseline persistence on this data proportion and calculated the prediction using root mean square error metrics.

<u>Baseline Model</u>

Here a simple baseline persistence model was carried out. The rmse value of the baseline model was 8.0

<u>Sarimax Model</u>

We used sarimax on the DMA-468 dataset. Performed grid search parameters using train set to get best parameters and using those we calculated weekly prediction with root mean square value 7.263 as the result.

<u>LSTM Model</u>

We used LSTM on the DMA-468 dataset. Here the look-back period of 3 days was considered to perform the weekly prediction. Hence finally achieved a root mean square value of 6.431 as the result.

<u>Plots of Baseline, Sarimax and LSTM - DMA 468 weekly</u>

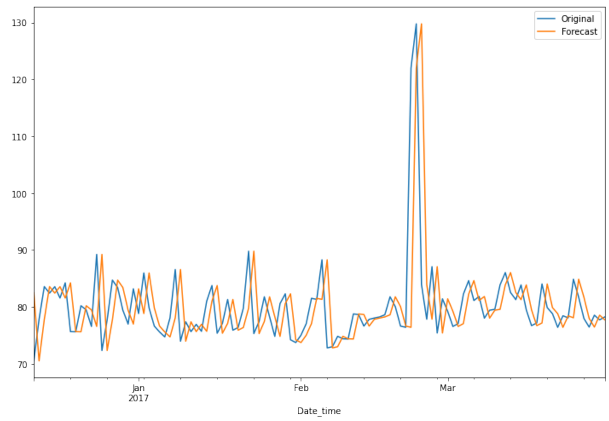

Figure 3.22. Baseline DMA468 univariate weekly prediction: original vs predicted plot

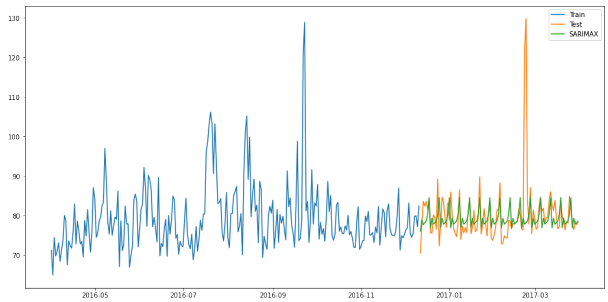

Figure 3.23. Sarimax DMA468 univariate weekly prediction

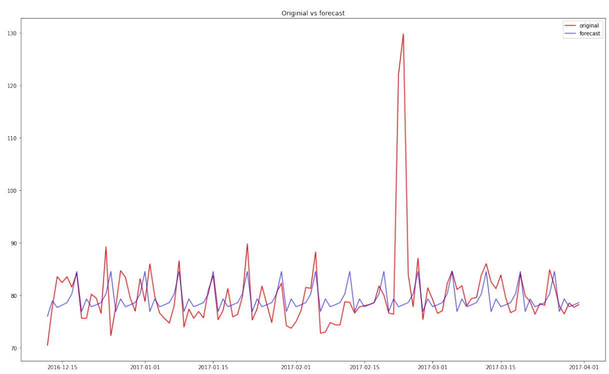

Figure 3.24. Sarimax DMA468 univariate weekly prediction: original vs predicted plot

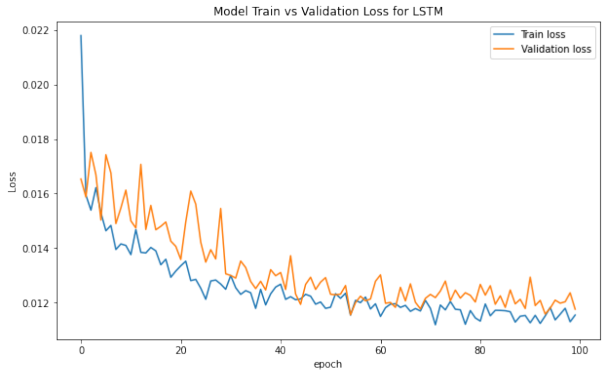

Figure 3.25. LSTM DMA468 univariate weekly prediction: lossgraph

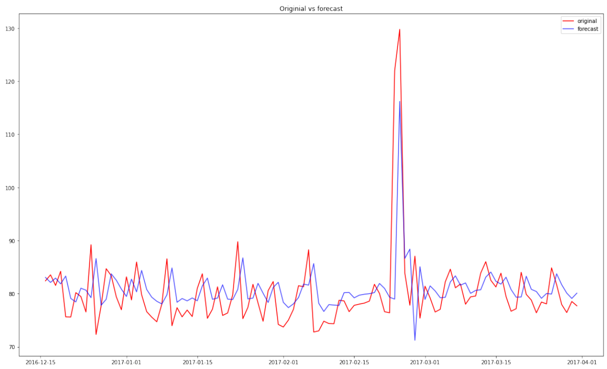

Figure 3.26. LSTM DMA468 univariate weekly prediction: original vs predicted plot

3.4.3 Univariate Time Series Analysis - monthly prediction

This includes prediction models implemented on two DMA's namely DMA 225 and DMA 468 respectively on a monthly basis. We performed prediction on the dataset, which was re-sampled on a daily basis. We performed models such as LSTM, sarimax and baseline.

DMA 225 Univariate Analysis: Monthly prediction

To perform univariate analysis on DMA 225 was re-sampled on a daily basis and the length of the dataset is 365. The last 3 months of the dataset was considered as a test set and the remaining was considered as a train set. We proposed LSTM and Sarimax and baseline persistence on this data proportion and calculated the prediction using root mean square error metrics.

Baseline Model

Here a simple baseline persistence model was carried out. The rmse value of the baseline model was 125.565.

Sarimax Model

We used sarimax on the DMA-225 dataset. We performed a grid search on a train set to get the parameters and use those parameters, we calculated monthly prediction with root mean square value 164.717 as the result.

LSTM Model

We used LSTM on the DMA-225 dataset. Here the look-back period of 27 days was considered to perform the monthly prediction. Hence we finally achieved a root mean square value of 21.772 as the result.

Plots of Sarimax, Baseline and LSTM - DMA 225 monthly

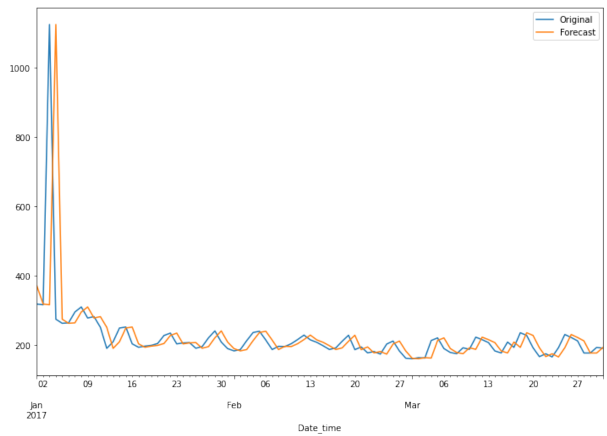

Figure 3.27. Baseline DMA225 univariate monthly prediction: original vs predicted plot

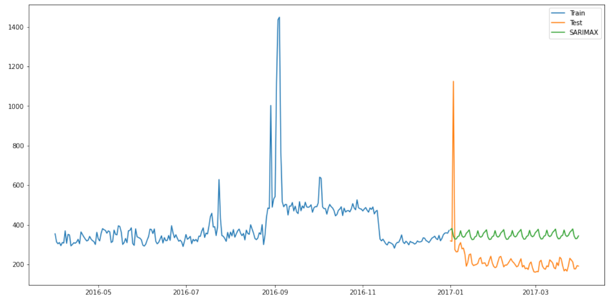

Figure 3.28. Sarimax DMA225 univariate monthly prediction

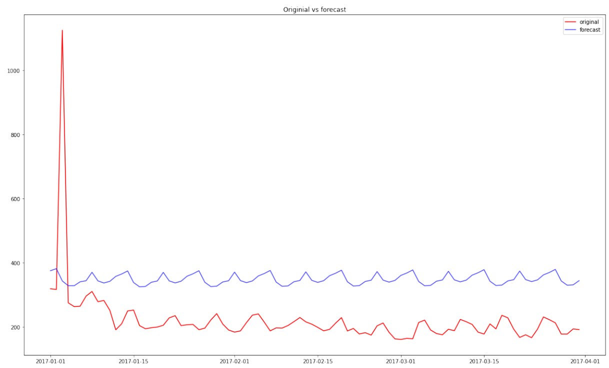

Figure 3.29. Sarimax DMA225 univariate monthly prediction: original vs predicted plot

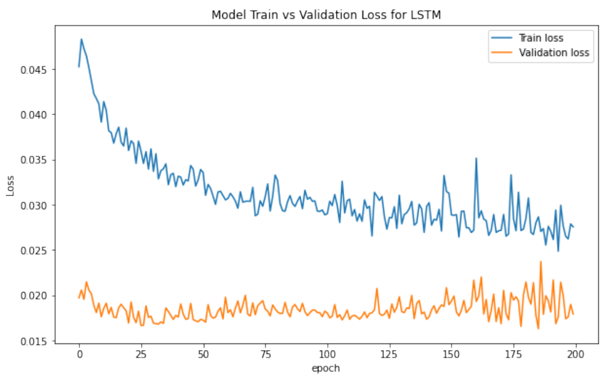

Figure 3.30. LSTM DMA225 univariate monthly prediction: lossgraph

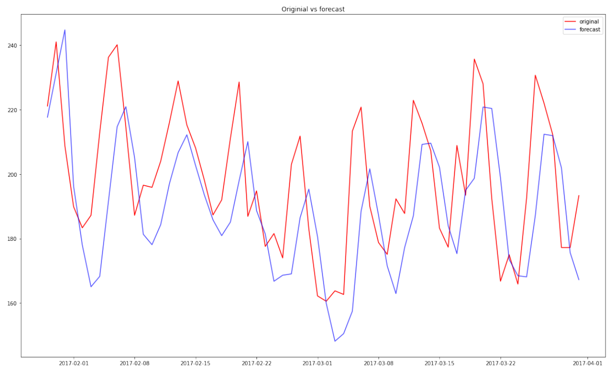

Figure 3.31. LSTM DMA225 univariate monthly prediction: original vs predicted plot

DMA 468 Univariate Analysis: Monthly prediction

To perform univariate analysis on DMA 468 was re-sampled on a daily basis and the length of the dataset was 365. The last 4 months of the dataset was considered as a test set and the remaining amount of data is considered as a train set. We proposed LSTM and Sarimax and baseline persistence on this data proportion and calculated the prediction using root mean square error metrics.

<u>Baseline Model</u>

Here a simple baseline persistence model was carried out. The rmse value of the baseline model was 7.796.

<u>Sarimax Model</u>

We used sarimax on the DMA-468 dataset. We performed a grid search on the train set to get the best parameters. Using those we calculated monthly predictions with a root mean square value of 7.39 as the result.

<u>LSTM Model</u>

We used LSTM on the DMA-468 dataset. Here the look-back period of 27 days was considered to perform the monthly prediction. Hence finally achieved a root mean square value of 6.836 as the result.

Plots of Baseline, Sarimax and LSTM - DMA 468 weekly

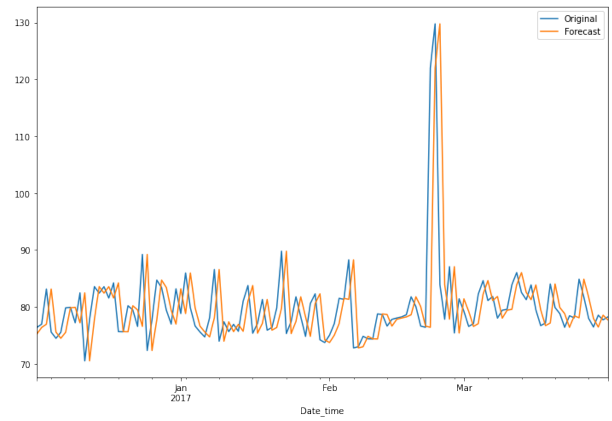

Figure 3.32. Baseline DMA468 univariate monthly prediction: original vs predicted plot

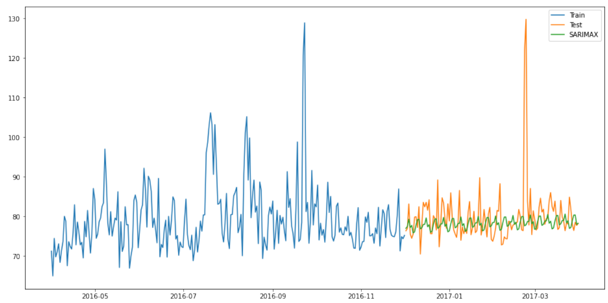

Figure 3.33. Sarimax DMA468 univariate monthly prediction

Figure 3.34. Sarimax DMA468 univariate monthly prediction: original vs predicted plot

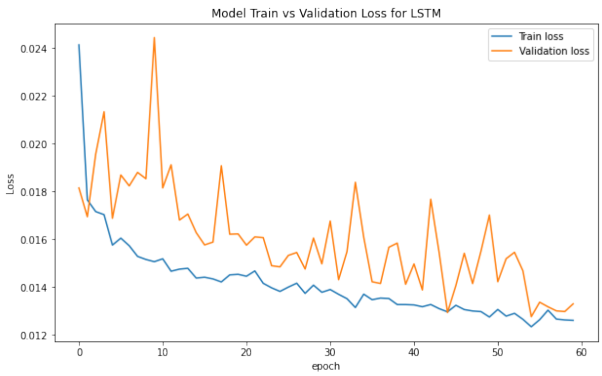

Figure 3.35. LSTM DMA468 univariate monthly prediction: original vs predicted plot

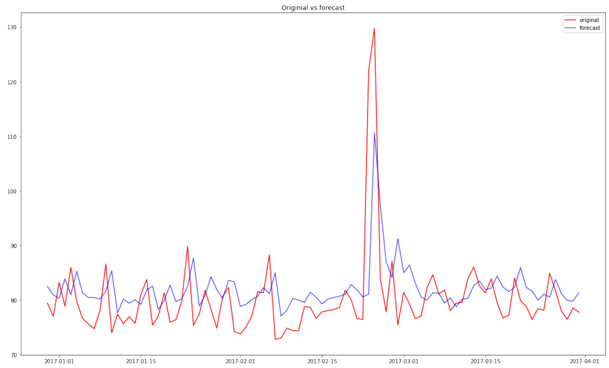

Figure 3.36. LSTM DMA468 univariate monthly prediction: original vs predicted plot

3.4.4 Multivariate Time Series Analysis - hourly prediction

This includes prediction models implemented on two DMA's namely DMA 225 and DMA 468 respectively on an hourly basis. Data was re-sampled on an hourly basis and predictions were carried out. Models such as LSTM, Facebook prophet were performed.

DMA 225 Multivariate Analysis: HOURLY

To perform univariate analysis on DMA 225, we considered the last 500 points as test set and the remaining portion of data to train set. We proposed LSTM and Facebook prophet, baseline persistence on this data proportion and calculated the prediction using root mean square error metrics.

<u>LSTM Model</u>

We used the TensorFlow library to perform the LSTM model on the DMA 225 dataset. The ultimate goal of this model was to calculate the hourly predictions. In the LSTM model, the look-back period was considered as 24 hours and output was predicted for 24 hours. So it followed a supervised learning problem approach. We obtained the rmse score of 0.835 as the result.

<u>Facebook Prophet Model</u>

We used the Facebook prophet library on the DMA-225 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 0.68 as the result.

Baseline persistence Model

We performed a baseline persistence model on the DMA-225 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 0.73 as the result

Plots of LSTM and Facebook prophet - DMA 225 hourly mv

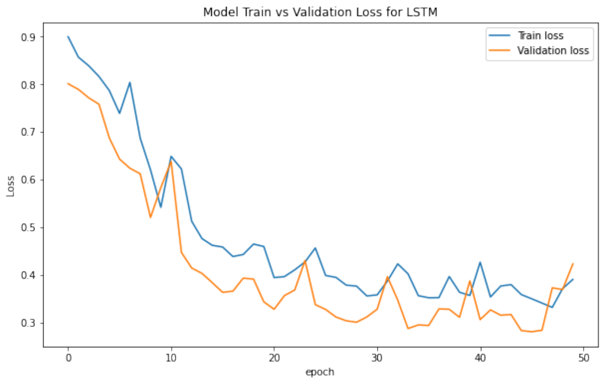

Figure 3.37. LSTM DMA225 multivariate hourly prediction: lossgraph

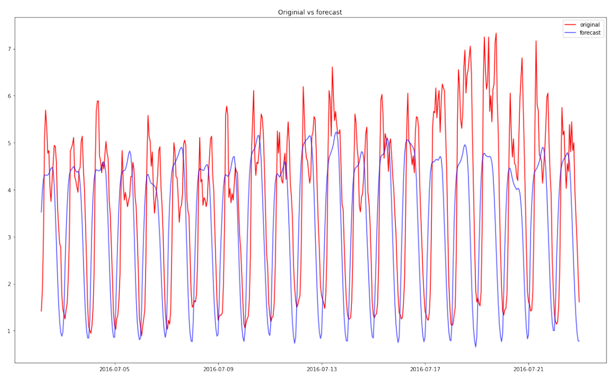

Figure 3.38. LSTM DMA225 multivariate hourly prediction: original vs predicted plot

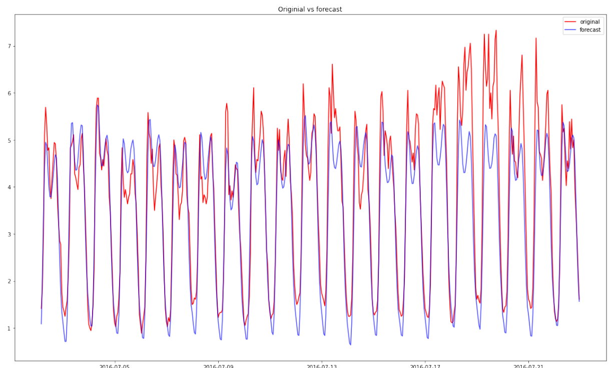

Figure 3.39. Facebook Prophet DMA225 multivariate hourly prediction: original vs predicted plot

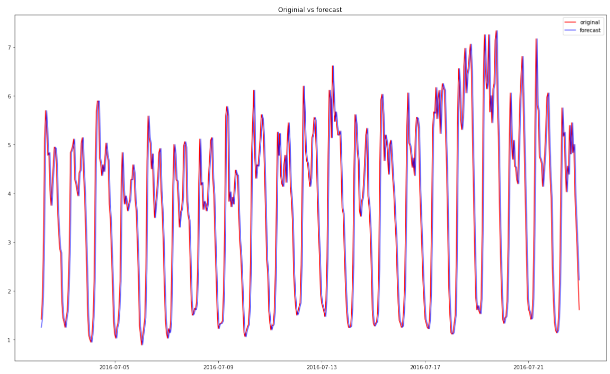

Figure 3.40. Baseline DMA225 multivariate hourly prediction: original vs predicted plot

DMA 468 Multivariate Analysis: HOURLY

To perform multivariate analysis on DMA 468, the last 442 data points were considered as test sets and the remaining were considered as a train set. We proposed the LSTM Facebook prophet and baseline persistence model on this data proportion and calculated the prediction using root mean square error metrics.

<u>LSTM Model</u>

We used the tensor flow library to perform the LSTM model on the DMA 468 dataset. The ultimate goal of this model was to calculate the hourly predictions. In the LSTM model, the look-back period was considered as 24 hours and output was predicted for 24 hours. So it followed a supervised learning problem approach. We obtained the rmse score of 0.196 as the result.

Facebook Prophet Model

We used the Facebook prophet library on the DMA-468 dataset. The ultimate goal of this model is to calculate the hourly predictions. We obtained the rmse score of 0.244 as the result.

Baseline Persistence Model

We performed a baseline persistence model on the DMA-468 dataset. The ultimate goal of this model was to calculate the hourly predictions. We obtained the rmse score of 0.239 as the result.

Plots of LSTM and Facebook prophet - DMA 468 hourly mv

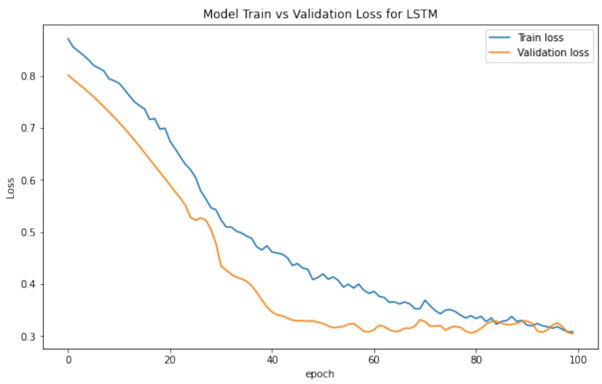

Figure 3.41. LSTM DMA468 multivariate hourly prediction: lossgraph

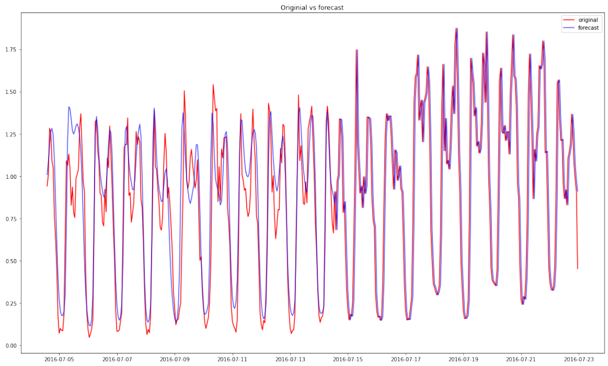

Figure 3.42. LSTM DMA468 multivariate hourly prediction: original vs predicted plot

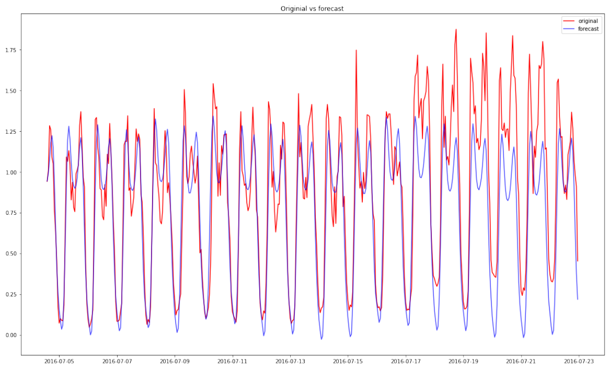

Figure 3.43. Facebook Prophet DMA468 multivariate hourly prediction: original vs predicted plot

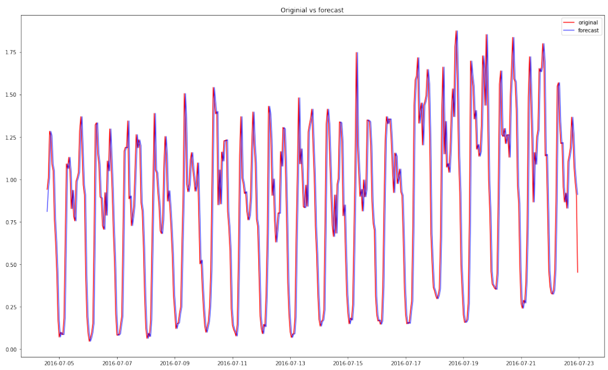

Figure 3.44. Baseline DMA468 multivariate hourly prediction: original vs predicted plot

Chapter 4

Experimental Results

This chapter describes the results which we achieved using our models. In a nutshell, we performed the below kinds of predictions.

(a) Hourly prediction - univariate and multivariate analysis
(b) Weekly prediction - univariate analysis
(c) Monthly prediction - univariate analysis

4.1 Hourly prediction

This subsection deals with the results of hourly prediction. We performed both univariate and multivariate analyses to achieve hourly prediction.

4.1.1 Univariate analysis: Hourly prediction

DMA	**Baseline**	**Fb prophet**	**LSTM**
DMA 225	2.156	4.227	1.436
DMA 468	1.084	1.2	0.779

Discussion of result: The above tabular column states that we considered two DMA's for hourly prediction and here it followed univariate analysis since the dataset had only the column of the target variable. We performed Baseline persistence, Facebook prophet and LSTM on the same train test proportion. Here root mean square error (RMSE/rmse) was considered as accuracy metrics. Our main goal was to compare the models fb prophet, LSTM with the baseline for each DMA's respectively. For univariate analysis LSTM performed better by achieving a lesser rmse score as 1.436 and 0.779 for DMA 225 and DMA 468 respectively. In figures 3.9 and 3.12 We achieved a good loss graph where training and validation loss merges resulting in a better rmse score with respect to lstm, which can be evident from figures 3.10 and 3.14 for both the DMA 225 and DMA 468 respectively. This is mainly lstm can handle time series or sequential data. So herewith LSTM we have the ability to decide and store the needed information with the help of gates

4.1.2 Multivariate analysis: Hourly prediction

DMA	Baseline	Fb prophet	LSTM
DMA 225	0.73	0.68	0.835
DMA 468	0.239	0.244	0.196

Discussion of result: The above tabular column states that we considered two DMA's for hourly prediction and here it followed multivariate analysis since the dataset had both dependent and independent variables. We performed Baseline persistence, Facebook prophet and LSTM on the same train-test proportion. Here root mean square error (RMSE/rmse) was considered as accuracy metrics Our main goal was to compare the models fb prophet, LSTM with the baseline for each DMA's respectively. For multivariate analysis of DMA 225, Facebook prophet performed better with the rmse score of 0.68, which was comparatively lesser than the baseline model. The main reason for that, was the DMA 225 had outliers and in our research, we tend to build the future prediction, without removing outliers. As we know Facebook prophet works best with time series which has strong seasonal effects and moreover it handles outliers well too. So we could achieve good results with the Facebook prophet as displayed in figure 3.39. Whereas with DMA 468 multivariate analysis, LSTM performed better by achieving a lesser rmse score of 0.196. In figure 3.41 we achieved a good loss graph where training and validation loss merges resulting in a better rmse score with respect to lstm, which can be evident from the prediction figure 3.42. The reason was that DMA 468, had consistent values for the entire time frame which we can relate to the weather measurements, recorded in the particular DMA found in HU1 locations, so this DMA 468 has fewer outliers comparatively and hence LSTM outperformed the other two models, as it does sequence modelling on the dataset.

4.2 Weekly prediction

This subsection deals with the results of the weekly predic tion. We performed univariate analysis to achieve weekly predictions.

4.2.1 Univariate analysis: Weekly prediction

DMA	Baseline	Sarimax	LSTM
DMA 225	113.697	132.542	91.412
DMA 468	8.057	7.263	6.431

Discussion of result: The above tabular column states that we considered two DMA's for weekly prediction and here it followed univariate analysis since the dataset had only the column of the target variable. We performed baseline persistence, Sarimax and LSTM models on same the train test proportion. Here root mean square error (RMSE/rmse) was considered as accuracy metrics. LSTM outperformed sarimax and baseline models, with lesser rmse scores of 91.412 and 6.431 for DMA 225 and DMA 468 respectively. With respect to DMA 225 and DMA 468, we could see the LSTM performance from figures 3.20, 3.21 and 3.25 and 3.26 respectively.

4.3 Monthly prediction

This subsection deals with the results of monthly prediction. We performed univariate analysis to achieve monthly prediction.

4.3.1 Univariate analysis: Monthly prediction

DMA	Baseline	Sarimax	LSTM
DMA 225	125.565	164.717	21.772
DMA 468	7.796	7.39	6.836

Discussion of result: The above tabular column states that we considered two DMA's for monthly prediction and here it followed univariate analysis since the dataset had the column of the target variable. We performed baseline persistence, Sarimax and LSTM models on the same train-test proportion. Here Root mean square error (RMSE/rmse) was considered as accuracy metrics. LSTM outperformed sarimax and baseline models, with lesser rmse scores as 21.772 and 6.836 for DMA 225 and DMA 468 respectively.

Chapter 5

Conclusions

This research project objective was to develop short term water demand forecasting using traditional and deep learning models. This kind of forecasting will help the organisation or client to plan and identify seasonality, annual patterns, production capacity, and expansion over the period based on requirements. This in turn contributes to the advent effect of business strategy(e.g., plans to launch a facility or store internationally and expand into new markets). Here we considered the weather dataset as an external factor and we figured out if it has an adverse impact on water demand forecasting on the particular date-time. In a nutshell, we performed hourly, weekly and monthly predictions using univariate and multivariate analysis strategies. Models such as LSTM(Long short-term memory), Fb prophet(Facebook prophet), Sari max (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors), Baseline(Persistence model) were carried out. LSTM outperformed the rest of the models, with respect to hourly, weekly and monthly predictions.

From the experiments and results, it's evident that deep learning models performed better than traditional models. We used tensor-flow and FB prophet library to carry out the short term water demand forecasting with deep learning techniques, whereas we used stats model sarimax library, to work with traditional models. The best model was decided based on rmse square value. Sarimax is a popular statistical method and it uses only past information to predict the future. Facebook prophet performed better with strong seasonal effects. It is also an additive model where non-linear trends are used. LSTM is a powerful algorithm derived from recurrent neural networks to perform time series forecasting. It has three gates and uses which LSTM manages to keep, forget, or remove data points. Additionally, LSTM also helps to overcome the vanishing gradient problem, once the prediction has been done, it is given inside the model as the input to predict the next output in the sequence. So basically activation functions are used to avoid exploding gradients. Compared to Sarimax, Facebook prophet models, LSTM performed better with lesser rmse values eventually in many use cases. The reason for lower rmse values is because of its capability to use previous sequential data, by this way the model has the ability to learn from long term observations of sequence data. This will pave the way for time series forecasting. When we have a dataset with long term seasonality(yearly pattern) or short term seasonality(weekly patterns), LSTM can be used. And one more good thing with lstm, is the tendency to accept inputs of different lengths and with that, it helps clients/ organisations to plan business strategy.

In a nutshell, to get an efficient prediction, LSTM can be implemented with deep learning, because of its properties to face the time series challenges and traditional models such as Sarimax are more efficient to produce good results for a smaller quantity of the dataset. In future research, we hope to implement and consider different models, to perform short term / long term water demand forecasting with some other external factors with respect to the water demand in each DMA. This will lead the client/organisation to plan business strategy in an effective way and achieve their final goals.

Bibliography

[1] Aileen Nielsen, *"Practical Time Series Analysis: Prediction with Statistics and Machine Learning"*, 2019 edition.

[2] Francesca Lazzeri, *"Machine Learning for Time Series Forecasting with Python"*, 2020 edition.

[3] Greg Rafferty,"Forecasting Time Series Data with Facebook Prophet", 2021 edition.

[4] Rob J Hyndman and George Athanasopoulos, *"Forecasting: Principles and Practice"*, 2nd edition.

[5] Bruce L. Bowerman, Richard O'Connell, Anne Koehler, *"Forecasting, Time Series, and Regression"*, 4th edition.

[6] Benjamin Kedem, Konstantinos Fokianos, *"Regression Models for Time Series Analysis"*, 1st Edition.

[7] František Štulajter, *"Predictions in Time Series Using Regression Models"*, 2002 edition.

[8] Robert Yaffee, Monnie McGee, *"An Introduction to Time Series Analysis and Forecasting"*, 1996 edition.

[9] Box, G. E. P., Jenkins, G. M. Reinsel, G. C. 1994 *"Time Series Analysis Forecasting and Control"*, 3rd edn. Prentice-Hall Englewood Cliffs, NJ.

[10] Stark, H.L., Stanley, S.J. and Buchanan, I.L, *" Water demand forecasting using artificial neural network"*, OWWA Conference paper, Ottawa, 1999.

[11] DeKay, C. Frederick, *"The Evolution of Water Demand Forecasting"*, 1985 edition.

[12] Bovas Abraham, Johannes Ledolter, *"Statistical Methods for Forecasting"*, 2009 edition.

[13] T. Kumura, N. Suzuki, M. Takahashi, S. Tominaga, S. Morioka, and S. Ivan, *"Smart water management technology with intelligent sensing and ICT for the integrated water systems"*, NEC Tech. J., vol. 9, no. 1,pp.103–106, 2015 edition.

[14] Ponte, B.; de la Fuente, D.; Pino, R. Rosillo, R. (2015): *"Real-Time Water Demand Forecasting System through an Agent-Based Architecture"*, International Journal of Bio-Inspired Computation, vol. 7, n. 3, (147-156).

[15] H. Park, *"Forecasting Three-Month Treasury Bills Using ARIMA and GARCH Models"*, Econ 930, Department of Economics, Kansas State University, 1999.

[16] Parrelli. R, *" Introduction to ARCH GARCH models", Optional TA Handouts*, Econ 472 Department of Economics, University of Illinois, 2002.

[17] J. Lee, *"Univariate time series modeling and forecasting (Box-Jenkins Method)*", Econ 413, lecture 4.

[18] K.W. Hipel, A.I McLeod,*"Time Series Modelling of Water Resources and Environmental Systems"*, 1994 edition.

[19] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, C. Muller, *"Neural modeling for time series: a statistical stepwise method for weight elimination"*, IEEE Trans. Neural Networks 6 (1995), pages: 1355–1364.

[20] Gyasi-agyei, Kwame, *"Analysis and modeling of prevalence of measles in the ashnti region of chana"*, Kwame Nkrumah University of Science and Technology, West Africa, 2012.

[21] Leonid Shvartser, Uri Shamir, Mordechai Feldman, *"Forecasting Hourly Water Demands by Pattern Recognition Approach"*Journal of Water Resources Planning and Management, volume 119, 1993.

[22] Ernesto Arandia, Amadou Ba, Bradley Eck, Sean McKenna S, *"Tailoring seasonal time series models to forecast short-term water demand"* Journal of Water Resources Planning and Management, volume 142, 2016.

[23] Di Tian, Christopher J. Martinez, Tirusew Asefa, *"Improving Short-Term Urban Water Demand Forecasts with Reforecast Analog Ensembles"* Journal of Water Resources Planning and Management, volume 142, 2016.

[24] Francesca Gagliardi, Stefano Alvisi, Zoran Kapelan, Marco Franchini, *"A Probabilistic Short-Term Water Demand Forecasting Model Based on the Markov Chain"*, Department of Engineering, University of Ferrara, Italy, 2017.

[25] Elena Pacchin, Stefano Alvisi, Marco Franchini, *"A Short-Term Water Demand Forecasting Model Using a Moving Window on Previously Observed Data"*, Department of Engineering, University of Ferrara, Italy, 2017.

[26] Billings. B, Jones. C, *"Forecasting Urban Water Demand"*, 2 nd edition, American Waterworks Association: Denver, CO, USA, 2008.

[27] Al-Zahrani, Abo-Monasar, *"Urban Residential Water Demand Prediction Based on Artificial Neural Networks and Time Series Models"*. Water Resource Management, 2015.

[28] S.L. Zhou, T.A. McMahon, A. Walton, J. Lewis, *"Forecasting operational demand for an urban water supply zone"*, Department of Civil and Environmental Engineering, Centre for Environmental Applied Hydrology, The University of Melbourne, Australia, 2001.

[29] A. Candelieria, F. Archetti, *" Identifying Typical Urban Water Demand Patterns for a Reliable Short-Term Forecasting – The Icewater Project Approach*, Department of Computer Science, System and Communication, University of Milano-Bicocca, Italy, 2014.

[30] M. Bakkera, H. Van Duist, K. Van Schagenb, J. Vreeburgd, L. Rietveld *" Improving the performance of water demand forecasting models by using weather input*, Delft University of Technology; Royal HaskoningDHV; PWN Waterleidingbedrijf Noord-Holland; Wageningen University; KWR Watercycle Research Institute Nieuwegein, The Netherlands, 2013.

[31] Jain, A., Varshney, K., Joshi, U., *" Short-term water demand forecast modelling at IIT Kanpur using artificial neural networks"*, IEE Transactions on Wate Resources Management 15 (1), 299–321, 2001.

[32] Zealand, C.M., Burn, D.H., Simonovic, S., *" Short term streamflow using artificial neural networks"*, Journal of Hydrology, 214, 32–48, 2005.

[33] John Bougadis, Kaz Adamowski2 and Roman Diduch, *" Short-term municipal water demand forecasting"*, Delcan Corporation, Ottawa; Department of Civil Engineering, University of Ottawa; Water and Wastewater Infrastructure, City of Ottawa, Ontario, Canada, 2005.

[34] Zhang, G., Qi, M., *" Neural network forecasting for seasonal and trend time series"*, European Journal of Operational Research 160, 501–514, 2005

[35] Shirley Gato, Niranjali Jayasuriya, Peter Roberts, *" Temperature and rainfall thresholds for base use urban water demand modelling"*, School of Civil, Environmental Chemical Engineering, RMIT University, Melbourne, Australia, 2007.

[36] Graeser Jr., H.J., *"Meter records in system planning"*, Journal of American Water Works Association 50 (11), 1395–1402, 1958.

[37] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, Michael Weyrich, *"A survey on long short-term memory networks for time series prediction"*, Procedia CIRP, Volume 99, Pages 650-655, 2021.

[38] Yupeng Wang, Shibing Zhu, Changqing Li, *Research on Multistep Time Series Prediction Based on LSTM"*, IEEE, 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), 2019.

[39] Wen-Xiang Fang, Po-Chao Lan, Wan-Rung Lin, Hsiao-Chen Chang, Hai-Yen Chang, Yi-Hsien Wang, *"Combine Facebook Prophet and LSTM with BPNN Forecasting financial markets: the Morgan Taiwan Index"*, IEEE, International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2019.

[40] Soukayna Mouatadid, Jan Adamowski, *"Using extreme learning machines for short-term urban water demand forecasting"*, Department of Bioresource Engineering, McGill University, QC, Canada, 2017.

[41] Zhang, Q., Diao, Y. and Dong, J., *" Regional water demand prediction and analysis based on Cobb-Douglas model"*, Water Resources Management, 27 (8), 3103–3113, 2013.

[42] Lippman RP, *"An introduction to computing with neural nets. IEEE ASSP Magazine"*, 4–22, 1987.

[43] Zurada, M. J, *" An Introduction to Artificial Neural Systems, PWS Publishing Company*, 1992. Mumbai, India.

[44] Msiza, I., Nelwamondo, F., Marwala, T., *"Artificial neural networks and support vector machines for water demand time series forecasting"*, IEEE International Conference on Systems, Man and Cybernetics, pp. 638–643, 2007.

[45] Durbin, J., Koopman, S., *"Time Series Analysis by State Space Methods"*, Oxford, 2001.

[46] Ashu Jain, Ashish Kumar Varshney, Umesh Chandra Joshi, *" Short-Term Water Demand Forecast Modelling Using Artificial Neural Networks"*, Department of Civil Engineering, Indian Institute of Technology, Kanpur, India, 2001.