Time Series Forecasting using various Machine Learning Models

by

VARUN SHAH
202011021

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

MAY, 2022

Declaration

I hereby declare that
(i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,
(ii) due acknowledgement has been made in the text to all the reference material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled Time series forecasting using various machine learning models has been carried out by Varun Shah (202011021) for the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology under my/our supervision.

Prof. Manjunath V Joshi
Thesis Supervisor

Acknowledgements

I would like to express gratitude to all of them who have supported me in my research work. This thesis is dedicated to Prof. Manjunath Joshi who was a major source of inspiration in guiding me throughout each and every possible direction in my research work.

Unfortunately, it wasn't possible to conduct physical meetings due to the pandemic, but then too he was always available for us for their guidance. I am thankful to him for giving me freedom in choosing appropriate methods to solve the given research problem, exploring the approach and then solving the issue by a new method altogether. I would also like to thank Prof. Avik Hati, and Prof. Shruti Bhilare for their constant efforts, comments and suggestions towards the methodology and technical work in regards to approaching a problem and its related implementation in real time scenario. I am thankful to my executive registrar and Dean AP who gave my campus visit permission so that I could conduct my research work as effectively as possible. I express my gratitude to Prof. Saurabh Tiwari under whom I did my teaching assistantship for three semesters for his support and guidance all through my journey at DAIICT.

I express my sincere thanks to my loving family and friends who were extremely helpful whenever there was a need. I would like to mention special thanks to Nidhi, Raj and Shantanu who were extremely helpful during all the important times, research that I conducted was not possible without their immense support. Thank you everyone for all your involvement.

Contents

Page No.

Abstract

Analysis of time series data is a challenging task in recent times. Statistical analysis of time series data and forecasting with the help of past data is a requirement in current times. The industry is looking forward to accomplishing complete effectiveness in forecasting. There are several established techniques such as auto regressing (AR), moving average (MA), autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) for univariate time series forecasting. For multivariate time series forecasting, the vector autoregression (VAR) model was used. With recent advances in deep learning techniques, prediction tasks can be effectively performed by a neural network and deep learning models can give better results than these established models. This study analyses and compares various established models with deep learning techniques on different datasets and explores whether transformers can be used for time series forecasting to get highly accurate results.

Keywords:- autoregression, transformer, forecasting, deep learning, statistical

List of Tables

List of Figures

Chapter 1
Introduction

Apart from the established techniques for forecasting time series, a recent trend is biased towards deep learning methods of solving the forecasting problem. As traditional methods still prove themselves as established and accurate, prediction using deep learning remains to be an unexplored problem. There are various research domains and huge applications of time series forecasting such as business, economics, engineering, politics, weather and forensics [1]. Time series analysis assists every type of organisation in understanding the primary reasons for foundational trends and patterns. Utilising time series data, business clients can see occasional patterns and delve further into why these patterns happen. The classification of data is either cross-section based or time series based. Cross-section data focuses on multiple variables at the same point in time for numerous subjects. Time series data focuses on variables of a particular subject at numerous time intervals. In time series, data must be collected at regular time intervals. In regression, we forecast the dependent variable from an independent variable, but in the time series, we are using the dependent variable only as the independent variable. Past observations are rigorously analysed and used to forecast future observations using these models or a neural network. Univariate time series data has one particular variable changing with respect to time. Multivariate time series data has numerous variables changing with respect to time. The most well-known method for univariate time series forecasting is the auto regressive integrated moving average (ARIMA) which is a combination of auto regressive (AR) and moving average (MA) [2]. Aniq et.al [3] explored the Vector autoregression (VAR) model for multivariate time series forecasting by examining causal relations between rainfall and temperature addressed by the VAR model. In recent times numerous deep learning methods like recurrent neural network (RNN) and long short term memory (LSTM) have gained much attention. With certain types of data, LSTM has been used in time series forecasting tasks [2]. A challenging field is analysing the accuracy of prediction when applying established techniques and deep learning based methods for time series forecasting. This thesis represents the literature review and corresponding implementation of the statistical methods as well as deep learning methods for forecasting future data.

1.1 Motivation

Time series analysis helps organisations understand the underlying causes of trends or systemic patterns over time. Policymakers and business supervisors on a standard premise use forecasts of financial variables to help make important choices about manufacturing, sales, demand, supply, market conditions, and other decisions regarding their fields. Time series analysis is a basic part of financial analysis with applications to the prediction of interest rates, unfamiliar money risk, stock market instability, and a parcel of other comparative undertakings and exercises. Economic time series are profoundly dependent and they correspond with other economic time series. Apart from financial and business domains, time series forecasting has huge applications in the medical domain and astronomy. Apart from traditional methods of forecasting time series data, deep learning methods are not explored up to the fullest. To the best of our knowledge, there is a need to find an optimal solution which can improve the forecasting results.

1.2 Problem Statement

This thesis focuses on forecasting univariate and multivariate time series data after analysing the previous observations and determining its model order and parameters subsequently minimising the forecasting error and comparing the state of the art methods with the deep learning methods. Multiple time series datasets are explored including stationary and non-stationary time series datasets to investigate various properties in a time series. Traditional methods, as well as deep learning based methods, are used and compared in order to get better forecasting results. A proposed method involves the use of transformers in time series forecasting.

1.2 Organisation of the Thesis

This thesis is composed of five chapters following the given sequence: Chapter 1 contains an introduction to time series data, along with the problem statement and motivation behind the research. The entire literature survey done within the timeline of this research including the traditional and statistical forecasting methods and deep learning methods is provided in chapter 2. The implementation details and its results for both univariate and multivariate datasets compared with both of these techniques are explained in chapter 3. The transformer model is proposed in chapter 4 which is used in multivariate forecasting analysis and the results are compared with deep learning based methods. Observations, discussions and results are the topics of chapter 5.

Chapter 2

Literature Survey

Time series analysis assists every type of organisation in understanding the primary reasons for foundational trends and patterns. There are several methodologies in time series forecasting which includes work done by soheila et al. in [1] by using improved ARIMA by applying mean of estimation error resulting in improved performance as compared to traditional ARIMA. Comparative features of time series forecasting are analysed by Mariia and Peter [4] by performing a few statistical experiments on feature based approaches which significantly depend on the characteristics of a time series. They also focus on optimal model selection criteria on the basis of trend and seasonality [4]. The authors in [5] conducted a review on the performance of various models for time series forecasting by exploring ARIMA, SARIMA, AR and Dynamic AR. The authors in [6] work upon forecasting the stock market price of SENSEX using a window technique of rapid miners. Traffic forecasting using time series data was performed by Shuvo [7] and others using ARIMA, SNAIVE and ETS models. According to [7] time series consists of trend, seasonality, cyclic trends, and irregularity. In the ARIMA model the I stands for integration, which makes sure that the time series is stationary ensuring that the mean and variance remains unchanged with time. Faraj et.al [8] compared the VAR model with the expectation minimization algorithm and prediction error minimization method and discussed its advantages and footfalls. Kartika and others [9] have used vector auto regression for estimation and modelling of stock price prediction and used genetic algorithms for further parameter optimization.

The authors in [10] show a comparative study of prediction algorithms like linear regression, support vector machines and multilayer perceptrons. Apart from these, there is some forecasting done by the deep learning methods as well which includes [11] i.e. forecasting using LSTM networks and then comparing the performance of ARIMA and LSTM on the basis of mean absolute percentage error (MAPE) evaluation metric. Anita et al. in [12] analysed stock price data which were extracted from ICICI, NIFTY and TCS and forecasting was done using stateful LSTM and stateless LSTM thereafter evaluating it with mean squared error (MSE). This paper [13] focuses on energy load forecasting for a comparatively smaller dataset and evaluates the results based on model order and parameter which gives the best RMSE value. The authors in [14] have tried the N-beats model for interpretable univariate time series forecasting on heterogeneous datasets.

2.1 Time series Analysis

Time series data focuses on perceptions of observations at various time intervals. Time series data should be gathered at multiple time intervals. In regression, say

$$Y = a + bX. \tag{2.1}$$

we forecast Y (dependent variable) from X (independent variable) where b represents the slope of the line and a represents the intercept (value of Y when X = 0), but in time series say Y_t is a time series such that,

$$Y_t = \beta Y_{t-1} + \epsilon_t, \tag{2.2}$$

where Y_{t-1} represents one time lag of time series Y_t, ϵ_t is the error term and β is the coefficient. Here we are using the dependent variable only as the independent variable unlike regression. Time series data focuses on observations of a particular subject at numerous time intervals. Data should be collected at multiple time intervals. Analysis of time series data is a challenging task as it has wide variations and randomness. Although some analysts have developed complex models evaluating the problem, recent advances have been made to solve those issues and forecast the future more accurately and precisely. A univariate time series data consists of a single variable varying with time, so univariate forecasting refers to predicting future data with the help of past observations.

For multivariate time series forecasting, a vector autoregression (VAR) model is utilised which fills in as an augmentation of the AR model [2]. The principal presumption for AR, MA and ARMA is that the time series is fixed. Assuming there are related trends and non direct seasonalities inside the time series then, at that point, forecasting error may increase and it can prompt incorrect and wasteful outcomes.

In time series analysis, investigators record data focused at steady intervals throughout a set timeframe. Data having arbitrary or random time intervals are not used in the process of forecasting. What separates time series data from different data is that the analysis can show how variables change over a long period. Time series data is represented in Fig 2.1 where the y-axis represents electric load and the x-axis is the timeframe in hours. The dataset consists of electric load on an hourly basis in MW units from 2004 to 2018 [21].

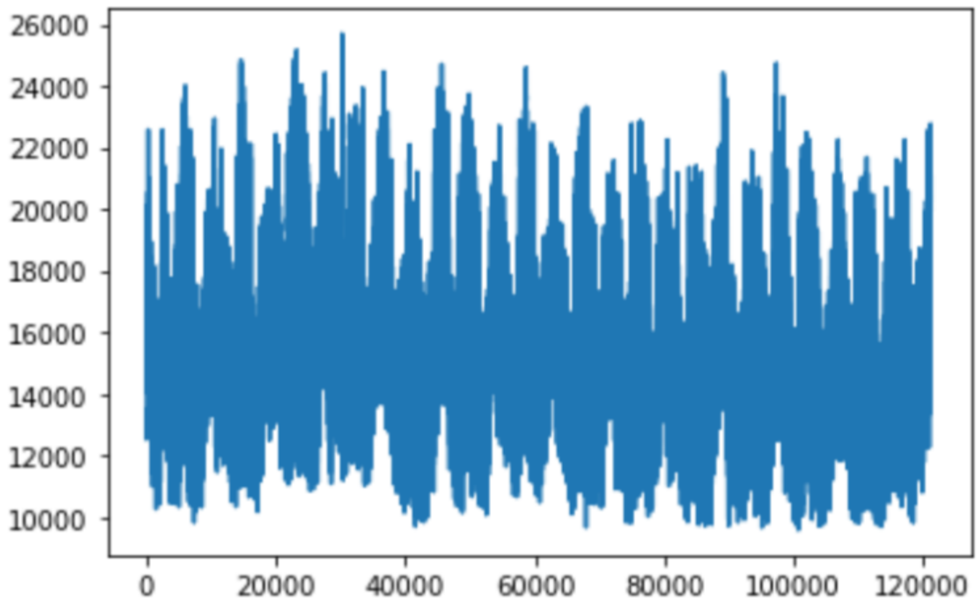

Fig 2.1 : Representation of time series data

2.2 Comparing performance of various models

Siami et.al [2] have compared ARIMA and LSTM models for time series forecasting and deduced that long short term memory (LSTM) models are superior to traditional and statistical time series forecasting models. The typical decrease in error rates of LSTM was between 84 - 87 percent when contrasted with ARIMA demonstrating the predominance of LSTM over ARIMA. Moreover, it was known that epochs had no major impact on the forecast model [2].

With the objective of examining the performance of traditional forecasting methods and deep learning-based methods, this research has analysed the performance of LSTM and ARIMA with regard to minimising the error rates in prediction. The review shows that the LSTM model outperforms ARIMA. LSTM is an extension to RNNs with extra features to remember the sequence of data [2].

The performance of LSTM and Bi-LSTM is compared by the same authors who report research on Bi-LSTM and LSTM models as shown in Fig 2.2. The results show that extra training of data and subsequently Bi-LSTM based models offer better predictions over normal LSTM based models. It was seen that Bi-LSTM models give better predictions when compared with ARIMA and LSTM models [16]. Fig 2.2 represents the LSTM and Bi-LSTM architecture where MLP stands for multi layer perceptron [2].

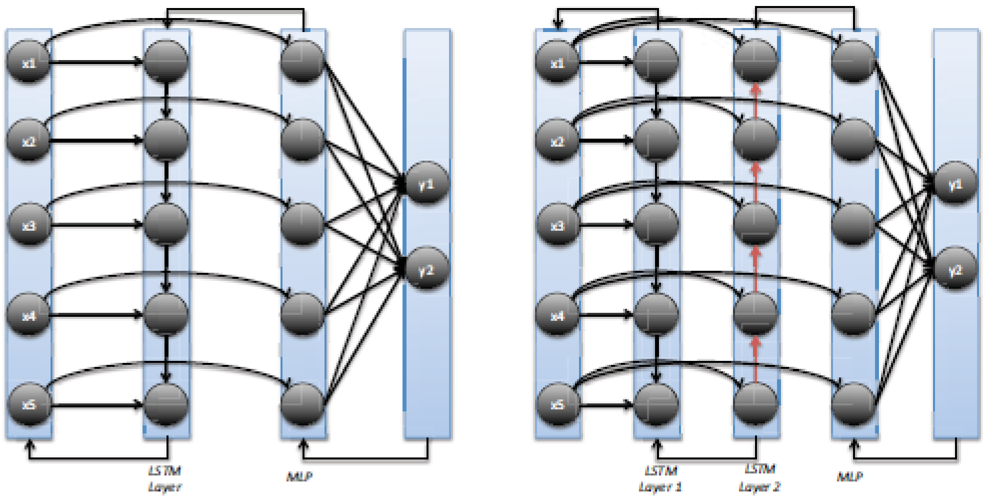

Fig 2.2: LSTM architecture vs Bi-LSTM architecture as in [2]

2.3 Univariate time series forecasting

2.3.1 Statistical models

Mariia et.al [4] have proposed a model comparison approach for univariate time series forecasting for all the statistical models like autoregression (AR), Moving average (MA), auto regressive moving average (ARMA) and auto regressive integrated moving average (ARIMA). A univariate time series data consists of a single variable varying with respect to time, so univariate forecasting refers to predicting future data with the help of past observations. A stationary time series is one whose statistical properties do not change or remain constant with time. In other words, time series with trends and seasonality are non stationary time series. In order to effectively forecast future observations, time series should be stationary. If not, the series could be converted from non stationary to stationary. ARIMA takes into account some constraints to convert non stationary time series data into stationary and then predicts the results. These models work for univariate time series data. The equations for these models are as follows:-

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots\ldots.. + \beta_p Y_{t-p} + \epsilon_t. \tag{2.3}$$

$$Y_t = \phi_0 + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots\ldots.. + \phi_q \epsilon_{t-q} + \epsilon_t. \tag{2.4}$$

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \epsilon_{t-1} + \ldots\ldots.. + \beta_p Y_{t-p} + \phi_q \epsilon_{t-q} + \epsilon_t. \tag{2.5}$$

Eq 2.3 addresses the auto regressive (AR) model, for example, if Y_t = {100,125,112,45,87……..} be a time series, then Y_{t-1} = {125,112,45,87……..} which represents one lag of Y_t , similarly Y_{t-2} represents two lag of Y_t . Y_{t-p} represents pth lag of Y_t t. The term ϵ_t is the white noise randomness called the error term. The term β_0, β_1,...., β_p represents the coefficients for time series lags represents Y_{t-1} , …., Y_{t-p} . Eq 2.4 addresses the moving average (MA) model formula with its coefficients ϕ_0,..... ϕ_q and Eq 2.5 represents combinations of both auto regressive and moving average models i.e. auto regressive moving average (ARMA) model. Error terms ϵ_t are assumed to be white noise with zero mean and some constant variance. Variance tells how far a dataset is spread out. It is mathematically defined as the average of the squared differences from the mean. Let S_t denote such a series which has $E\ \{S_t\} = 0$, where E stands for Expectation and constant variance $V\ \{S_t\}\ =\ \boldsymbol{\sigma}^2$ and auto-covariance = 0 i.e. each observation should be uncorrelated with other observations in the sequence, that is called the white noise process in a time series i.e Series which is purely random in nature, so prediction is not feasible for white noise processes. In the AR model Y_t depends particularly on its past values Y_{t-1} , Y_{t-2} with its associated coefficients β_0, β_0 etc. Y_{t-1} is one lag of Y_t . AR model depending on p number of passed values is known as AR (p), where p is called the model order parameter. In moving average Y_t is represented as the linear combination of random error terms with its corresponding coefficients depending on q of its past error terms known as MA (q). When a time series is considered as a linear combination of both AR and MA models it is called ARMA (p, q). ARIMA is an extension of the ARMA model introducing a term called integrated which indicates the stationarity feature of a time series. These are the steps involved while forecasting with the help of time series.

AR, MA and ARMA models have certain limitations, if the time series data is non-stationary it is seen that they would not give good forecasting results. So deciding whether a time series is stationary or not becomes an important factor to achieve better forecasting results.

2.3.2 Identifying stationarity in a time series

A time series Y_t is said to be stationary if the mean, variance and covariance of a series are time-invariant. Covariance refers to the relationship between two variables when a particular variable changes. If a decrease in one variable results in a decrease in the other variable, both these variables are said to have positive covariance. Whether the series is stationary or not can be checked with the help of an ADF (Augmented dicker fuller) test. Once the series is stationary we can apply any of these AR, MA or ARMA models and forecast the future observations. If the series is processed without converting it into stationary, then established techniques might give inaccurate forecasting results. These models work well only when the series is stationary.

Time series are stationary in the event that they don't have a trend or seasonal impact. At the point when a time series is stationary, it tends to be more straightforward to model. Observations from a non-stationary time series show seasonal impacts, trends, and different designs that rely upon the time data. Classical time series analysis and forecasting techniques are concerned with making non-stationary time series data stationary by distinguishing and eliminating trends and eliminating seasonal impacts.

If a series is non-stationary then there is a process to convert a series into stationary, which is done with the help of differencing. Differencing helps remove trends and seasonality present in the data. Trend refers to an increase or decrease in a particular observation over time. Seasonality refers to variation in a time period of data at a fixed interval. Both need to be removed in order to make a time series stationary. Performing differencing over log transformation or power transformation gives us the difference form of data.

In order to check the stationarity of a time series data, Let y_t be a time series,

$$y_t = 3(y_{t-1}) - 2.75(y_{t-2}) + 0.75(y_{t-3}) + u_t \tag{2.6}$$

3, -2.75, 0.75 be the coefficient values obtained for respective lags $y_{t-1}, y_{t-2}, y_{t-3}$ and u_t is the error term. To know whether this is a stationary series or not the process is as follows:- Let,

$$y_{t-1} = L(y_t), \tag{2.7}$$

$$y_{t-2} = L^2(y_t), \tag{2.8}$$

$$y_{t-3} = L^3(y_t), \tag{2.9}$$

Substituting it in Eq 1.6, we get

$$y_t - 3(y_{t-1}) + 2.75(y_{t-2}) - 0.75(y_{t-3}) - u_t = 0 \tag{2.10}$$

$$y_t\,(1 - 3L + 2.75L^2 - 0.75L^3) = u_t \qquad (2.11)$$

Characteristic equation would be as follows:-

$$1 - 3z + 2.75z^2 - 0.75z^3 = 0, \qquad (2.12)$$

After factoring,

$$(1-z)\,(1-1.5z)\,(1-0.5z) = 0, \qquad (2.13)$$

Hence, The value of z is 1, ⅔, 2.

Series is said to be stationary when all the roots are greater than 1. Thus, this is a non-stationary series. This is the method through which we can check the stationarity of a time series.

2.3.3 Limitations of ARIMA

Parameters (p, d, q) where p is the model order for AR, q is the model order for MA and d is the number of times differencing needs to be done in order to convert non stationary series into stationary, need to be manually defined, so finding the most accurate fit can be a long trial-and-error process. Additionally, the model depends highly on the reliability of historical data. It depends on most previous lags in the forecasting process. It is difficult to predict the turning points for ARIMA. It is computationally expensive and there is poor performance for long term forecasts. It handles non-stationary data well but still cannot be compared with deep learning based forecasting results. ARIMA cannot be used for seasonal time series.

2.4 Multivariate time series forecasting

2.4.1 Vector Autoregression model

Vector Autoregression (VAR) is a multivariate forecasting method that can be used when two or additional time series impact one another i.e. the connection between the time series included is bi-directional. It is considered as an autoregressive model as every variable of a time series is modelled as an element of the past observations i.e. the lags (time delay) of the series. The difference between AR, MA, ARMA and ARIMA to those of the VAR model is that they are unidirectional, where the impact on the series is in one way and not the other way around. Though, Vector auto regression (VAR) is bi-directional when the variables impact one another.

Let Y_t be a time series of GDP per year and X_t be a time series for per capita income. Let's assume both are interdependent on each other. VAR(1) model i.e. VAR model of order 1 is defined as shown in Eq 2.14 and Eq 2.15.

$$Y_t = A_1 + C_{11}Y_{t-1} + C_{12}X_{t-1} + \epsilon_t. \quad (2.14)$$

$$X_t = A_2 + C_{21}X_{t-1} + C_{22}X_{t-1} + \epsilon_t. \quad (2.15)$$

In Eq 2.14 and 2.15, terms A_1, A_2, C_{11}, C_{22}, C_{12}, C_{22}, refers to the coefficients for the time lags Y_{t-1}, X_{t-1}. ϵ_t is the error term.

2.4.2 Multivariate forecasting using Deep learning based methods

When it comes to deep learning based methods, long short term memory (LSTM) is a widely used method in time series forecasting. LSTM networks were planned explicitly to overcome the dependency issue looked at by recurrent neural networks RNNs (because of the vanishing gradient and diminishing gradient problem). LSTMs have feedback connections which make them different to feedforward neural networks. This property empowers LSTMs to handle whole successions of data like time series.

LSTMs utilise a series of gates which control how the information in a succession of data comes inside the network. There are three gates in LSTM called a forget gate, input gate and output gate. These gates can be considered as filters.

Bidirectional LSTM which is an extension to LSTM has also been experimented in order to forecast future samples. Bi-LSTM relies on the past data to predict the future, but it also executes learning from future observations to predict the past observations. Regular LSTM can make input flow in a particular direction either backwards or forward. In Bi-LSTM we can preserve past as well as future information.

2.5 Time series forecasting using Transformers

Transformer is a multi-head attention based state of the art deep learning model which works on encoder-decoder based architecture [15]. Transformers can also be used in order to forecast time series samples. Transformers must generate a forecasting sequence of observations along the time axis. Transformers architecture is based on self-attention mechanisms. An attention function can be defined as mapping a query and a bunch of key-value pairs to an output, where all queries and key-value pairs along with the output are vectors. The weighted sum of corresponding values is the output and the respective weight assigned to each value is calculated by a function of the query with its respective key as shown in Fig 2.3.

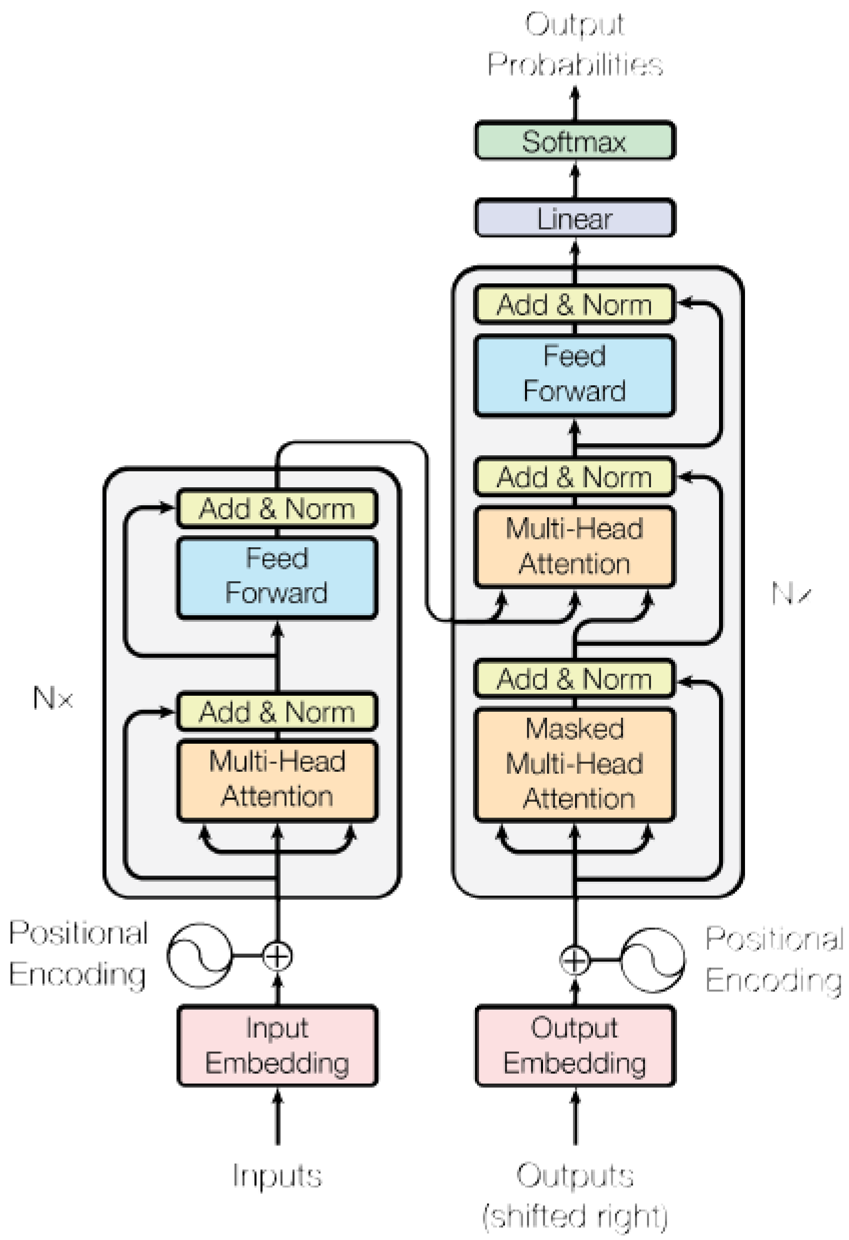

Fig 2.3: Transformer architecture by jones et.al [15]

Self-attention, also known as intra-attention is an attention mechanism utilised effectively in many applications including understanding appreciation, abstractive summarization, textual entailment and learning task-autonomous sentence representations [15].

The encoder is made out of a stack of N = 6 indistinguishable layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a position-wise fully connected feed-forward network.

The decoder is made out of a stack of N = 6 indistinguishable layers. In addition to 2 sub-layers in each encoder layer, the decoder embeds a third sub-layer, which performs multi-head attention over the result of the encoder stack as shown in Fig 2.3. Like the encoder, They employ residual connections around every one of the sub-layer, followed by layer normalization.

2.6 Chapter Summary

In this chapter the detailed literature survey which includes univariate and multivariate time series analysis techniques along with traditional and deep learning based techniques for time series forecasting was discussed. Time series data properties like stationarity and how to convert a non stationary series into stationary and then applying statistical models for prediction were detailed in this chapter. The role of transformers in time series forecasting was also included.

Chapter 3

Work Done: Comparison of performance of univariate and multivariate time series forecasting

In this chapter, univariate and multivariate time series forecasting is performed over traditional forecasting techniques and the results are compared with deep learning based forecasting techniques. The univariate time series forecasting process is as follows:-

3.1 Univariate time series forecasting

Fig 3.1 shows the flowchart of the time series forecasting process. Firstly there is a need to check whether the series is stationary or not. If yes then the process of differencing can be followed multiple times as per the need. Then through ACF and PACF plot model and its order can be determined, followed by forecasting the variable and calculating the root mean square error (RMSE).

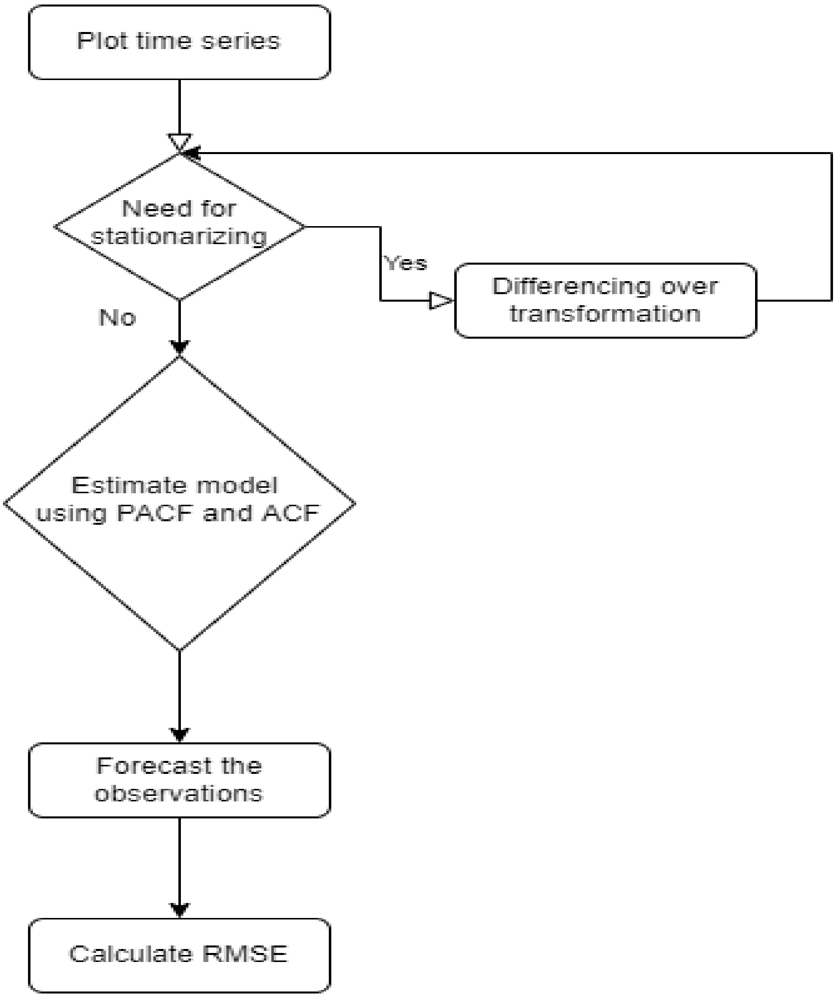

Fig 3.1: Time series forecasting process

Various experiments have been performed on different datasets by varying training and testing ratios, varying the size of the datasets and analysing the results obtained. The terms inside the bracket in Table 3.1 indicates the value of model order p and q for AR and MA respectively. The value beside it indicates the minimum value of root mean square error (RMSE) for Table 3.1. Due to limitations in the ARIMA model several hit and trial experiments have been done by exploring different datasets [22] and then comparing the results which were performed on the original paper to ensure proper accuracy in these results. The results are as shown in table 3.1 for 8 arbitrary data for all classical models.

The process of differencing helps reduce the variation between consecutive data points, which eventually helps in maintaining the uniformity of data. As shown in Fig 3.3 the differenced data when compared over original data, an upward increasing trend originally present in data is now removed after the process of differencing only once. The series achieves stationarity by performing differencing over transformation once or multiple times.

Table 3.1

Univariate forecasting results with various datasets [22]

Dataset List	AR	MA	ARMA	ARIMA
Female birth	(1,0) 4.12	(0,1) 5.14	(1,1) 4.14	(1,0,1) 4.14
Shampoo sales	(3,0) 221.22	(0,2) 278.61	(2,1) 206.74	(2,1,1) 164.87
Beer productions	(2,0) 23.40	(0,2) 24.13	(2,1) 24.76	(2,1,1) 32.59
Spot Prize	(2,0) 12.36	(0,1) 13.56	(2,1) 12.35	(2,0,1) 12.35
Monthly sunspots	(2,0) 55.01	(0,1) 65.91	(2,1) 53.10	(2,0,1) 53.10
Consumption data	(2,0) 88.15	(0,1) 102.39	-	-
Load hourly	(1,0) 604.16	-	-	-

3.1.1 Converting non-stationary data to stationary

For explaining the process of converting a non-stationary time series into stationary series, air passengers data [23] was used which is a non-stationary dataset. The process of converting a non-stationary series into stationary series can be achieved through log transformation followed by lag-1 differencing. The other way to remove trends and seasonality is through decomposition. In the first method, after applying log transformation to the original data it is subtracted from lag-1 of log transformation as shown in Fig 3.3, where you can see the mean and variance being constant over time. We can also use power transformation over the log transformation for the same process. A series which becomes stationary after differencing once is said to be integrated of the order of 1. The process can be repeated multiple times till the series is stationary. Thus in ARIMA (p, d, q), d is the differencing order which indicates how many times differencing is done in order to make a series stationary. We can apply differencing multiple times if the series is not obtained to be stationary after performing the differencing process once. It is confirmed that after performing the process multiple times, the value of mean and variance would be constant and the series would convert into a stationary series.

Eq 2.6, 2.7. 2.8, 2.9, 2.10, 2.11, 2.12, 2.13 can be referred to which shows the maths behind checking whether a time series is stationary or not. It can be seen in the representation of data in Fig 3.2 where there is an increasing trend. When trend and seasonality change in data is detected then there is a need to convert it into stationary. Decomposition is another inbuilt python function that helps to separate the residuals and trends from the underlying series.

The correct order of differencing is the minimum differencing expected to get a near-stationary series which revolves around a characterised mean and the ACF plot compasses to zero genuinely fast. To eliminate trend (increase or decrease in time-series value over time) and seasonality (variations in measured value which repeats over the same time interval regularly) we can apply log transformation, power transformation, differencing or both of these operations. Assuming the autocorrelations are positive for some number of lags (at least 10), then, at that point, the series needs further differencing. In Fig 3.2 and Fig 3.3 the y-axis represents the passengers attending the flight over a period of three months and the x-axis represents the timeframe.

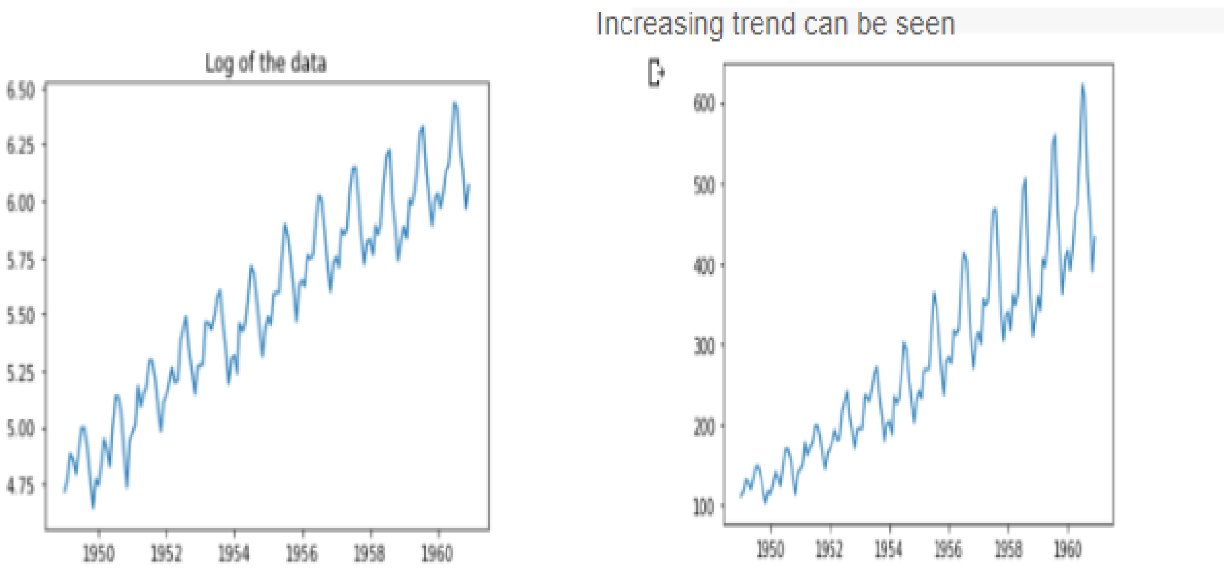

Fig 3.2: Log of air passenger data vs Original air passenger data [23]

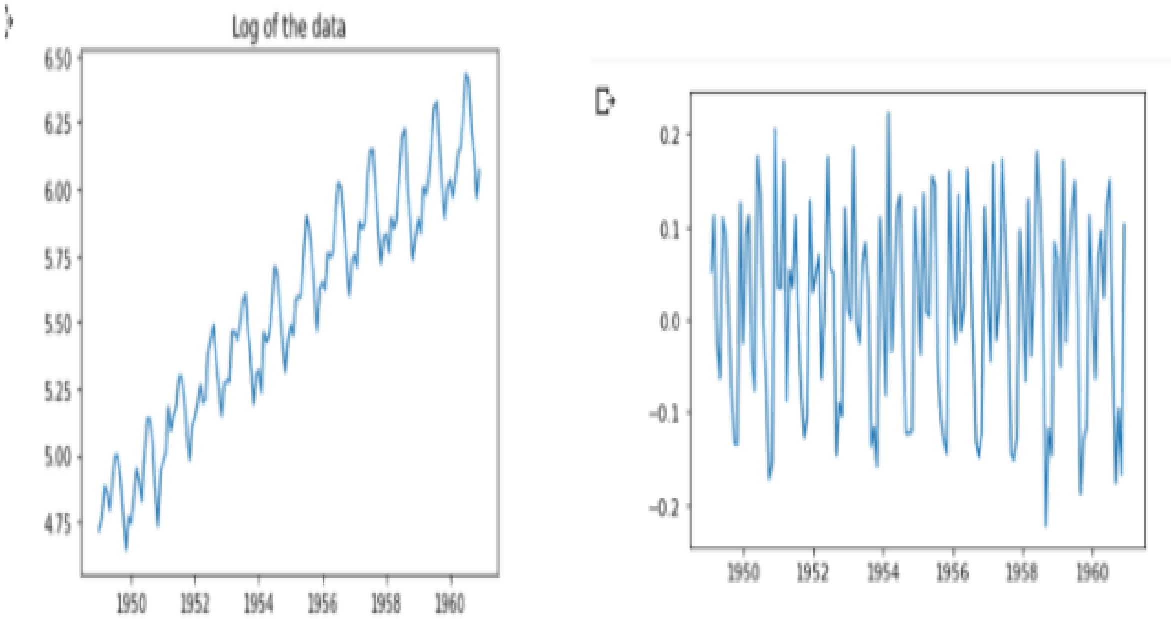

Fig 3.3: Log of air passenger data [23] vs Lag-1 Differenced data

3.1.2 Estimation and forecasting of univariate time series data

Estimation and forecasting of a univariate time series is carried out by the Box-Jenkins method [11]. It consists of :

1) Identification
2) Estimation
3) Diagnostic Checking

Under Identification there are two functions used to determine the order of AR and MA i.e. the value of p and q. Those functions are :

1) Autocorrelation function (ACF)
2) Partial Autocorrelation function (PACF)

Autocorrelation function (ACF) refers to the way the observations in a time series are related to each other. It is measured by a simple correlation between current observation Y_t and observation p periods from the current one Y_{t-p}. It helps to determine the order of MA models.

$$ϱ = Correl\,(Y_t\ ,Y_{t-p}) = 1/N \sum_{t=1}^{n-p} ((Y_t - \overline{Y})(Y_{t-p} - \overline{Y})\ /\ (Y_t - \overline{Y})^2) \tag{3.1}$$

where ϱ is the auto correlation function (ACF). Eq 3.1 states the mathematical formula for ACF. ACF considers how observation in a time series is related to each other. It is measured by a single correlation between Y_t and observation p periods from the current i.e. Y_{t-p}. PACF measures associated degrees between Y_t and Y_{t-p} when effects of another time lags are removed. The equation for PACF is as follows for time lags i.e. $p = 1$.

$$ψ = Correl\,[(Y_1, Y_0)] \tag{3.2}$$

$$ψ = Correl\,[(Y_k, Y_p^{p-1}, Y_0, Y_0^{p-1})] \tag{3.3}$$

Here in Eq 3.2 and Eq 3.3, ψ represents PACF function and p refers to time lags. If the value of $p >= 2$, PACF function is as shown in Eq 3.3.

The main question that emerges is why there is a need to calculate ACF and PACF. The plot of ACF, PACF and lags help us determine the model order p and q. The value of ACF at respective lags can be manually determined whether the value of p should be 1,2,3 etc similarly for q. PACF measures the degree of association between Y_t and Y_{t-p} when the effect of other time lags are removed. The comparison of correlograms (Plot of sample ACF vs lags) of time series data with theoretical ACF and PACF leads to the selection of appropriate models (AR, MA, ARMA). PACF plot is used in determining the order of AR. The plot for PACF and ACF is shown below in Fig 3.5 and Fig 3.4 respectively. As you can observe it mostly depends on the previous 3 lags which can be represented as the linear combination of the previous 3 time lags of the auto regressive model as PACF is used to determine the value of p that is the order of AR. In Fig 3.4 and Fig 3.5 X-axis represents the time lags and Y-axis represents the value of ACF or PACF obtained respectively.

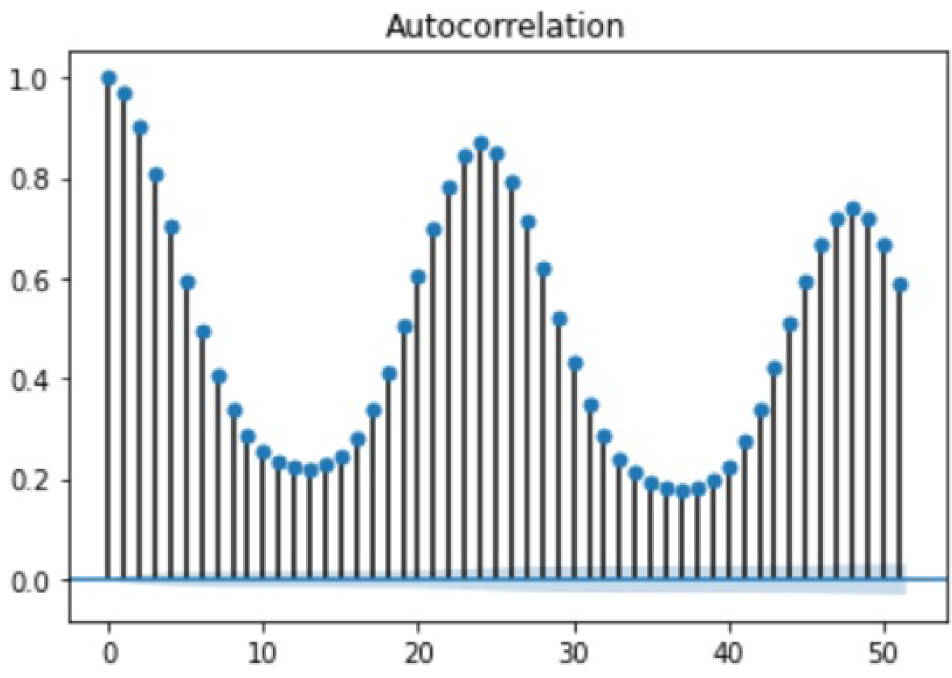

Fig 3.4: ACF curve

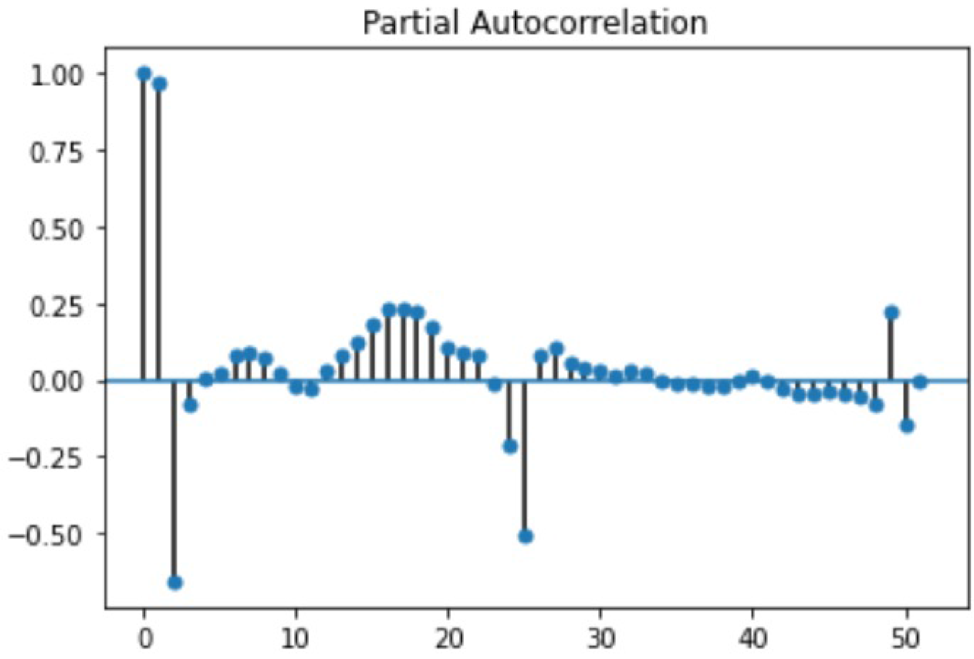

Fig 3.5: PACF plot

Fig 3.4 shows the plot of the autocorrelation function which is used to determine the order of MA i.e value of q. For estimation and diagnostic checking, there are several methods used to estimate the parameters of these models such as the yule walker procedure, maximum likelihood estimation, moments method etc. For diagnostic checking, various tests are performed on the model.

3.1.3 Determining model order and parameters

There is a lot of hit and trial involved in order to determine model order and then using mean squared error as an evaluation metric, an appropriate model can be chosen giving the lowest RMSE value as given in Fig 3.5. The datasets were divided into an 80:20 ratio for training and testing while evaluating the results for load forecasting. Although there are certain limitations when it comes to ARIMA models which can work only when the time series is stationary [10]. Lots of experiments need to be performed in order to prepare accurate predictions, as shown in Fig 3.6, there are several values of p and q called as model order is determined manually. Nowadays there is some scope for deep learning based techniques to solve the problem of forecasting in time series data. LSTM and Bi-LSTM techniques are implemented on the same dataset to check the accuracy of forecasting and compare it with traditional multivariate forecasting models.

Table 3.2

Comparing LSTM and ARIMA for univariate time series forecasting [22]

Dataset List	ARIMA	LSTM
Female birth	(1,0,1) 4.14	3.30
Shampoo sales	(2,1,1) 164.87	217.47
Beer productions	(2,1,1) 32.59	19.58
Spot Prize	(2,0,1) 12.35	0.51
Monthly sunspots	(2,0,1) 53.10	20.78

Deep learning based LSTM methods are compared with traditional models on the same set of datasets to observe the forecasting performance on all types of stationary and non stationary data. Experimental results proved to be in the favour of LSTM based methods in comparison with the established ones. In a few cases, it was observed that for smaller non-stationary time series data, sometimes ARIMA gave slightly better results than LSTM. But in the majority of the cases, LSTM proved better than both ARIMA and VAR models too. The model orders have been determined by extensive hit and trial to obtain as minimum root mean square error (RMSE) as possible. By analysing the ACF and PACF plots for univariate time series, model order was determined and forecasting was performed.

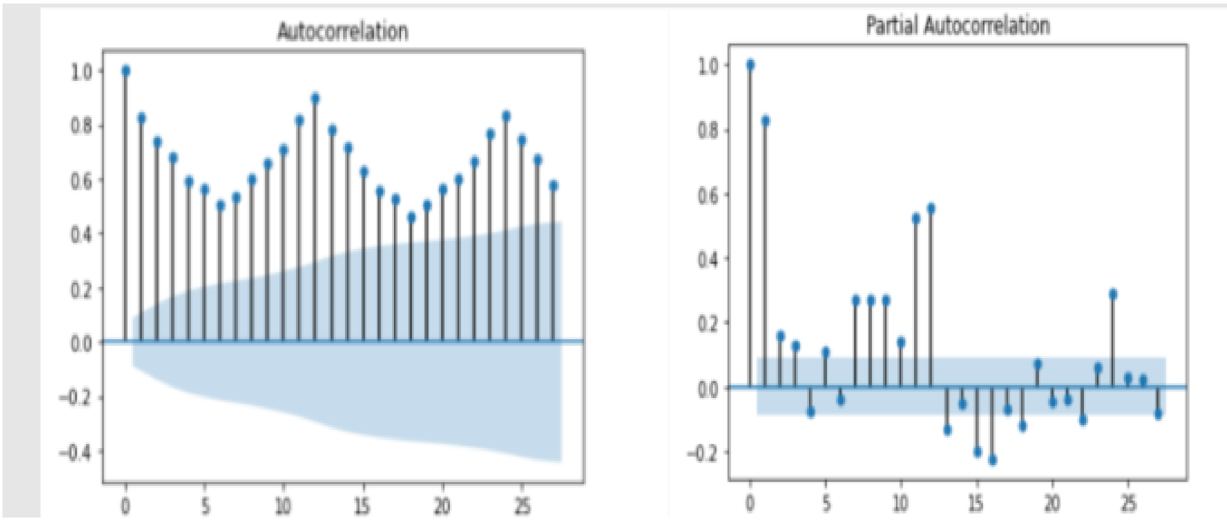

Fig 3.6: ACF and PACF plot for monthly beer dataset [24]

Fig 3.6 shows ACF and PACF plots for the monthly beer dataset, as PACF is used to determine the order of AR. From Fig 3.6, it can be made out that AR(2) can be suitable as you can see that dependency is on the first two lags. So using AR(2) can give the best forecasting results.

The loss function for all univariate time series models is mean squared error (MSE). In this Fig 3.7, you can see the predicted results with minimum forecasting error after extensive hits and trials involved in model configurations.

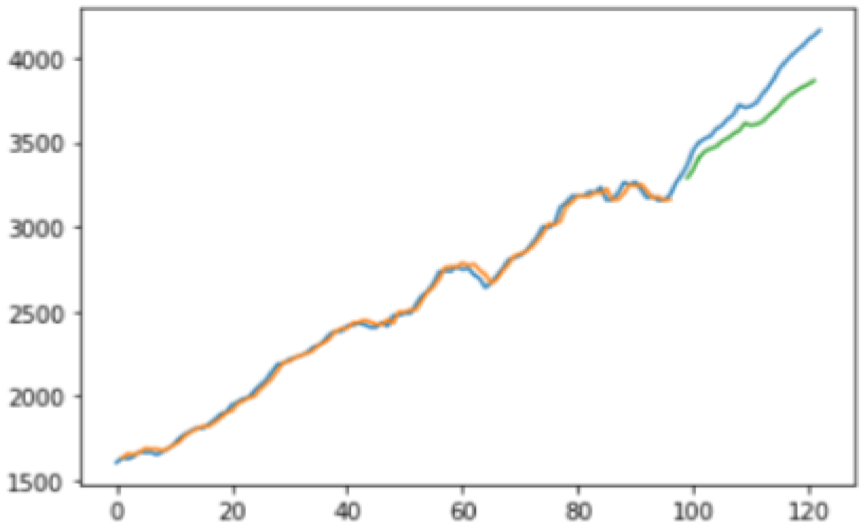

Fig 3.7: Actual vs Predicted results

3.2 Multivariate time series forecasting

In multivariate time series forecasting there would be more than one variable varying with respect to time and the task of forecasting would be to forecast a particular variable's future values based on its past values as well as the past values for all the variables. There would be multiple time series corresponding to each and every observation.

3.2.1 Multivariate forecasting using Vector-Autoregression (VAR)

For multivariate time series forecasting, the Gross national product dataset [26] is used which has 8 different time series namely unit labour cost (ULC), real gross national product (RGNP), fixed weight deflator (GDF), fixed weight deflator for food (GDFCF) etc [26]. Experiments have been performed to check the interdependency between all of them and prediction results are verified thereafter. The task is to forecast one with the help of another time series i.e. multivariate time series forecasting with the Vector Autoregression (VAR) model. The loss function in the VAR model is mean square error function.

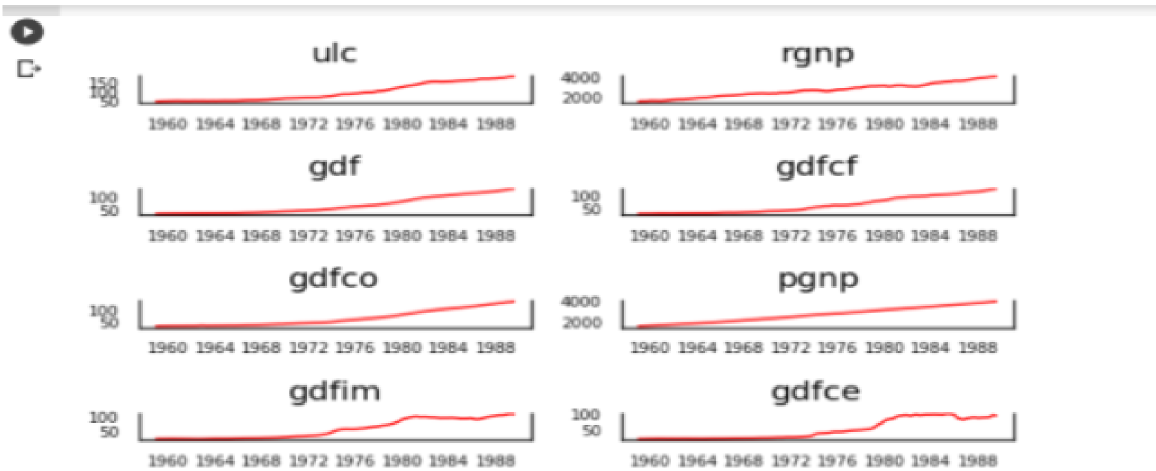

Fig 3.8: Plotting GNP multivariate dataset [26] of 8 time series

Experiments have been done with all 8 time series but as the trends show, there was no major interdependency between all eight of them with each other. So results would differ and there would be some best case amongst 3,4,5,6,7,8. The best case was obtained to be of 4 time series namely ulc, rgnp, gdf and gdfcf, which had the most interdependence between each other. This table shows the respective mean value and mean square error for each prediction result, the order was fixed to 3 for each case. In Fig 3.8 X-axis represents the timeframe and Y-axis represents the unit labour cost (ULC) or real gross national product (RGNP) for each time instant respectively.

Table 3.3

VAR multivariate forecasting best case

Time series	Mean Value	RMSE
ULC (Unit Labour cost)	178.56	2.514
RGNP (real gross national product)	3970.69	238.54
GDF (fixed weight deflator)	122.99	4.238
GDFCF (fixed weight deflator for food)	121.475	5.10

3.2.2 Multivariate forecasting using LSTM

For multivariate time series forecasting using deep learning based methods i.e. LSTM, which involves data preprocessing and training the model. The overall strategy is to clean, scale, and split the data prior to making the tf.dataset object.

Cleaning data: Filled any missing qualities with a linear interpolation whenever necessary.

Scaling data: Data is scaled whenever required.

Data splitting and model configuration: The train and validation sets are made with an 80:20 split and loss plots across the models are genuinely steady. The LSTM starts to turn out to be very overfit from about epoch 100 where the validation loss starts to rise. The lstm_skip likewise has a point around epoch 50 where the val_loss quits diminishing. In all cases, this is a sign when models are done learning against the validation set.

The forecasting execution of utilising a few different model types has been done here where each model proposes a similar last two DNN layers with dropout. One of 128 units, and the last layer of 24 (output horizon). Every one of the model's different layers is

A three-layer DNN (one layer in addition to the base two layers)
A CNN with two layers of 1D convolutions with max pooling.
An LSTM with two LSTM layers.
A CNN stacked LSTM with layers from models 2 and 3 into the DNN layer.
A CNN stacked LSTM with a skip connection to the DNN layer.

Mean square error is the loss function used in LSTM, which is the mean overseen data of squared difference between actual and forecasted observations. In Eq 3.2 y is the actual value and y_i is the predicted value and N is the number of training examples.

$$L(y, y_i) = 1/N \sum_{i=1}^{N} (y - y_i)2 \qquad (3.2)$$

3.2.3 Multivariate forecasting using Bi-LSTM

Bidirectional long short term memory (Bi-LSTM) is the most common way of making any neural network which has the sequence information in the two headings backward (future to past) and forward (past to future).

In bidirectional, our input flows in two headings, making a Bi-lstm not the same as the regular LSTM. With the regular LSTM, we can make an input stream in one heading, either backwards or forward. Nonetheless, in bi-directional LSTM, the input stream can be made in both the headings to preserve future and past information. In the pollution dataset [26], the task is to predict the pollution at the current hour given the pollution measurement and weather conditions (Temp, pressure, dew point etc) at the prior time step. There are 7 input variables and 1 output variable here. Bi-LSTM serves as a better approach for multivariate time series forecasting as compared to the vector autoregression (VAR) model and LSTM. Mean squared error is the loss function used in Bi-LSTM as shown in the Eq 3.2.

Table 3.4 shows the comparison results between VAR, LSTM and Bi-LSTM where Bi-LSTM outperforms the other two. Note that the results are obtained after extensive changes in model configurations so it may not be always possible for every multivariate time series dataset to give better forecasting results in Bi-LSTM as compared to LSTM. The result behind Bi-LSTM giving better results than LSTM is that Bi-LSTMs have extended the capabilities of LSTM by training the input data in both forward and backward directions. Thus it helps the prediction model to obtain better accuracy and experiments have proven its robustness on different datasets.

Table 3.4

Comparing VAR, LSTM, Bi-LSTM for multivariate time series forecasting

Dataset	VAR	LSTM	Bi-LSTM
Gross National Product	2.514	1.32	1.174
Pollution	56.14	23.86	0.27

3.3 Chapter Summary

In this chapter, univariate time series forecasting results are compared over 8 datasets for AR, MA, ARMA, ARIMA and LSTM models. Additionally for multivariate time series forecasting traditional and deep learning based methods results were compared with VAR, LSTM and Bi-LSTM models for these datasets to ensure uniformity in prediction results.

Chapter 4

Work Done: Time series forecasting using Transformers

Transformer is a multi-head attention based state of the art deep learning model which works on encoder-decoder based architecture [15]. Transformers are originally used in solving natural language processing (NLP) type of problems which involve semantics and context mainly used in language translation. Here the aim is to use the transformer model in multivariate time series forecasting. The research is based on whether transformers can be used in time series forecasting and how better they would be able to predict future observations.

4.1 Multivariate time series forecasting using transformers

Transformers can also be used in order to forecast time series samples. Transformers must generate a forecasting sequence of observations along the time axis. Transformers architecture is based on self-attention mechanisms. An attention function can be defined as mapping a query and a bunch of key-value pairs to an output, where all queries and key-value pairs along with the output are vectors. A weighted sum of corresponding values is the output and the respective weight assigned to each value is calculated by a function of the query with its respective key.

In this research, the transformer is used as the accurate method of forecasting time series data and not in language translation or NLP which is the traditional job of a transformer. A neural network forecasting model named as N-Beats called Neural Basis Expansion Analysis for Time Series is used in order to forecast the future observation[14]. A multi forecasting library named darts is used which combines forecast related modules of Pytorch and several other modules. Dart also helps switching between forecast methods and data preprocessing. Dart adopts the traditional N-Beats architecture to multivariate time series by flattening the source data to a uni-dimensional series. Energy Dataset is used which consists of hourly records of electricity price levels between 2015 and 2018, in Euros per megawatt-hour, energy demand in MWh, energy generation type coal, gas etc [25].

4.2 N-BEATS Transformers model

The task is to forecast the price levels by considering all these 29 variables which influence these price levels. Thereafter a probabilistic forecast would be derived using the quantile regression technique i.e. neural network's loss function can be regulated as a quantile loss function. Quantile regression will calculate the central forecasting value at each given time step. While training the N-Beats transformer model, parameter likelihood is set to QuantileRegression having parameters as quantiles which executes the model to draw respective samples from a quantile loss function at each particular time step. This will create a probabilistic forecast of quantiles instead of a mere pointwise estimate. For a set of predictions, the loss will be its average. Using the quantile loss function, the main advantage is that quantile loss provides sensible prediction intervals even for residuals with non-constant variance or non-normal distribution. The transformer served as a better approach than the LSTM for the energy dataset as shown in table 4.1.

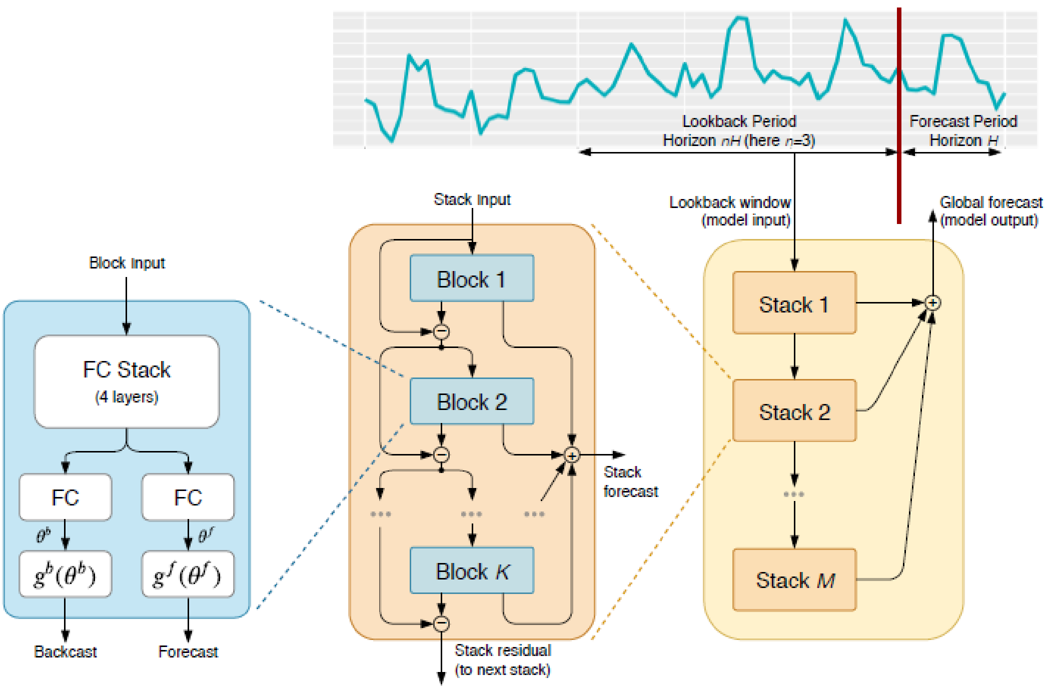

Fig 4.1: N-BEATS model architecture by Boris et.al [14]

A quantile regression loss function is applied to forecast quantiles. A quantile is a value below which a fraction of observation in a group falls. Eq 4.1 shows the quantile loss function equation. Given a prediction y_i^{p} and outcome y_i, the mean regression loss for a quantile q whose value lies between 0 to 1 is as follows:-

$$L(y_i^{p}, y_i) = max[q(y_i^{p} - y_i), (q - 1)(y_i^{p} - y_i)] \tag{4.1}$$

Boris et.al [14] has originally used N-Beats architecture for time series forecasting, They have used a set of loss functions to build an ensemble for accurate forecasting. They have used MAPE (Mean Absolute Percentage Error) as an evaluation metric. Their MAPE value is 18.52.

4.3 Comparing Results

Energy Dataset is used which consists of hourly records of electricity price levels between 2015 and 2018 [25], in Euros per megawatt-hour, energy demand in MWh, energy generation type coal, gas etc. Transformer has the capacity to fit long input sequences. The encoder-decoder architecture of the transformer processes the input sequences and predicts the value of each and every variable at future timesteps as an individual token. Thereafter the prediction error metric can be minimised and the model creates a range of forecasts. The transformer served as a better approach than the LSTM for the energy dataset as shown in table 4.1.

Table 4.1
Comparing LSTM, transformer for multivariate forecasting

Dataset	LSTM	N-Beats Transformer
Energy	450.79	5.47

Table 4.2 compares traditional methods and deep learning based methods with transformers for multivariate forecasting on GNP and Energy dataset respectively. The results as shown in table 4.2 states that the transformer model stands out in forecasting performance as compared to the traditional methods.

Table 4.2
Comparing other methods with transformer for multivariate forecasting

Dataset	VAR	LSTM	Bi-LSTM	Transformer
Gross National Product	2.514	1.32	1.174	0.98
Energy	1114.6	450.79	402.63	5.47

4.3 Chapter Summary

In this chapter, the proposed method in which the N-BEATS transformer model is used for multivariate time series forecasting. The forecasting result is thereby compared with LSTM forecasting results which showed considerable improvement over LSTM in RMSE value.

Chapter 5
Discussions and Conclusion

Discussion of Results

Deep learning based techniques like LSTM perform forecasting on a time series much better as compared to traditional forecasting techniques like ARIMA and VAR. While comparing LSTM with Bi-LSTM on a particular dataset, Bi-LSTM gives better results. But that cannot be said for each and every dataset as this involves a lot of manual changes in model configurations like changes in batch size, learning rate and epochs. Using a transformer model for multivariate time series forecasting gave extraordinary results as compared to LSTM. Experimentations have been done on two datasets where transformer model stands out in forecasting results as compared to deep learning models.

The iterative optimization algorithm for deep learning based methods involves a lot of training parameters which eventually helps in training the model in a better way positively affecting the prediction results. By manually varying batch size, no. of iterations and epochs better results are obtained in deep learning based methods like LSTM and Bi-LSTM. In some cases the accuracy of the model worsened than the traditional model resulting in overfitting of the model. Hence the results in Table 3.2 and Table 3.4 are displayed after performing proper trails of all these parameters on the respective datasets.

Conclusion

This effort was an attempt of forecasting time series data along with comparing different time series data on different model orders and parameters. LSTM based architecture can be proposed that can solve the problem of accurate prediction and furthermore increase the performance of forecasting [10]. ARIMA model gives good results in univariate time series forecasting but future work focuses on deep learning based methods. LSTM has feedback connections unlike any other neural network, here the model would learn about the function that maps the sequence of observations. LSTM and Bi-LSTM proved to be better in both univariate and multivariate cases. The implemented components are verified by numerous datasets [22][23][24][25] for both univariate and multivariate cases. The transformer was used to forecast multivariate time series on one of the datasets and got better results as compared to LSTM too. A transformer is normally used in language translation and NLP, but it can also be used in place of deep learning techniques for time series forecasting.

References

[1] S. Mehrmolaei and M. R. Keyvanpour, “Time series forecasting using improved arima,” in 2016 Artificial Intelligence and Robotics (IRANOPEN), 2016, pp. 92–97.

[2] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A comparison of arima and lstm in forecasting time series,” in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394–1401.

[3] A. A. Rohmawati and P. Gunawan, “The causality effect on vector autoregressive model: The case for rainfall forecasting,” in 2019 7th International Conference on Information and Communication Technology (ICoICT), 2019, pp. 1–5.

[4] M. Mariia and G. Peter, “Model selection approach for time series forecasting,” in 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), 2019, pp. 1–5.

[5] Priyamvada and R.Wadhvani, “Review on various models for time series forecasting,” in 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 405–410.

[6] U. Agarwal and A. S. Sabitha, “Time series forecasting of stock market index,” in 2016 1st India International Conference on Information Processing (IICIP), 2016, pp. 1–6.

[7] M. A. R. Shuvo, M. Zubair, A. T. Purnota, S. Hossain, and M. I. Hossain, “Traffic forecasting using time-series analysis,” in 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 269–274.

[8] F. Bashir and H.-L. Wei, “Handling missing data in multivariate time series using a vector autoregressive model based imputation (var-im) algorithm: Part i: Var-im algorithm versus traditional methods,” in 2016 24th Mediterranean Conference on Control and Automation (MED), 2016, pp. 611–616.

[9] F. Kristianda, Irhamah, and K. Fithriasari, “Estimation of vector autoregressive model’s parameter using genetic algorithm,” in 2018 International Symposium on Advanced Intelligent Informatics (SAIN), 2018, pp. 72–77.

[10] O. Meryem, J. Ismail, and E. M. Mohammed, “A comparative study of predictive algorithms for time series forecasting,” in 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 68–73.

[11] M. S. Hossain and H. Mahmood, “Short-term load forecasting using an lstm neural network,” in 2020 IEEE Power and Energy Conference at Illinois (PECI), 2020, pp. 1–6.

[12] A. Yadav, C. K. Jha, and A. Sharan, “Optimizing lstm for time series prediction in indian stock market,” Procedia Computer Science, vol. 167, pp. 2091–2100, 2020, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920307237

[13] F. Razak, M. Shitan, A. Hashim, and I. Zainal Abidin, “Load forecasting using time series models,” Jurnal Kejuruteraan, vol. 21, 12 2009.

[14] B. Oreshkin, D. Carpo, N. Chapados, and Y. Bengio, “N-beats: Neural basis expansion analysis for interpretable time series forecasting,” 05 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc.

[16] S. Siami-Namini, N. Tavakoli and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285-3292, doi: 10.1109/BigData47090.2019.9005997.

[17] A. A. Rohmawati and P. H. Gunawan, "The Causality Effect on Vector Autoregressive Model: The Case for Rainfall Forecasting," *2019 7th International Conference on Information and Communication Technology (ICoICT)*, 2019, pp. 1-5, doi: 10.1109/ICoICT.2019.8835379.

[18] A. K. Biswas, S. I. Ahmed, T. Bankefa, P. Ranganathan and H. Salehfar, "Performance Analysis of Short and Mid-Term Wind Power Prediction using ARIMA and Hybrid Models," *2021 IEEE Power and Energy Conference at Illinois (PECI)*, 2021, pp. 1-7, doi: 10.1109/PECI51586.2021.9435209.

[19] N. Kumar and S. Susan, "COVID-19 Pandemic Prediction using Time Series Forecasting Models," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225319.

[20] T. Jakaša, I. Andročec and P. Sprčić, "Electricity price forecasting — ARIMA model approach," *2011 8th International Conference on the European Energy Market (EEM)*, 2011, pp. 222-225, doi: 10.1109/EEM.2011.5953012.

[21] J. Contreras, R. Espinola, F. J. Nogales and A. J. Conejo, "ARIMA models to predict next-day electricity prices," in IEEE Transactions on Power Systems, vol. 18, no. 3, pp. 1014-1020, Aug. 2003, doi: 10.1109/TPWRS.2002.804943.

[22] S. A. Abbas et al., "Performance Analysis of Classification Algorithms on Birth Dataset," in IEEE Access, vol. 8, pp. 102146-102154, 2020, doi: 10.1109/ACCESS.2020.2999899.

[23] B. N. K. Sai and T. Sasikala, "Forecasting Air Passenger Data using various models," 2019 Third International Conference on Inventive Systems and Control (ICISC), 2019, pp. 95-100, doi: 10.1109/ICISC44355.2019.9036333.

[24] Godahewa, Rakshitha & Bergmeir, Christoph & Webb, Geoffrey & Hyndman, Rob & Montero-Manso, Pablo. (2021). Monash Time Series Forecasting Archive.

[25] Mel Keytingan M. Shapi, Nor Azuana Ramli, Lilik J. Awalin, Energy consumption prediction by using machine learning for smart building: Case study in Malaysia, Developments in the Built Environment, Volume 5, 2021, 100037, ISSN2666-1659,https://doi.org/10.1016/j.dibe.2020.100037.
(https://www.sciencedirect.com/science/article/pii/S266616592030034X)

[26] Han, Lingyi & Peng, Yuexing & Li, Yonghui & Yong, Binbin & Zhou, Qingguo & Shu, Lei. (2018). Enhanced Deep Networks for Short-Term and Medium-Term Load Forecasting. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2888978.