Forecasting Foreign Tourist Arrival in India: Neural Network vs. Time Series Models

Project report submitted in partial fulfillment of the requirement for

POST-GRADUATE DIPLOMA IN
STATISTICAL METHODS AND ANALYTICS

submitted by

Anurima Dey & Shreyasee Dev
Roll no: (DST-19/20-005) & (DST-19/20-021)

27 July, 2020

Indian Statististical Institute, North-East Centre, Punioni, Solmara, Tezpur 784501, Assam

Certificate

This is to certify that Anurima Dey and Shreyasee Dev have done the project under my supervision (from 02/2020 to 07/2020). This is an original project report based on work carried out by them in partial fulfillment of the requirement for the Post-Graduate Diploma in Statistical Methods and Analytics programme of the Indian Statistical Institute, North-East Centre, Tezpur, Assam.

(Supervisor's signature)

Kushal Banik Chowdhury

Acknowledgements

We feel lucky to be guided by Dr. Kushal Banik Chowdhury and also for his encouragement, advice, and care. Our zeal for learning Time Series has just begun under such guidance, for which we shall be forever grateful to him. Without his support and guidance, we would not have been able to complete this project. We also appreciate his way of questioning, conceptualizing, and thinking, which made us focus onto more detailed understanding, while never forgetting to look into the data handling part of the analysis. We would like to express our heartiest thanks to Dr. Sanjit Maitra for being there as a constant support. We would like to express our heartfelt gratitude and respect to Prof. Nrityananda Sarkar, the previous Head of the NE branch of ISI, Tezpur for his guidance from time to time. We thank all other faculty members for their encouragement and assistance. We shall remain ever indebted to our parents for always motivating and supporting us in every aspect of our life and career.

Date: 27/July/2020

Anurima Dey Shreyasee Dev

Anurima Dey & Shreyasee Dev

Contents

Chapter 1

Introduction

At this age of globalization, the Travel and Tourism Industry(T&T) plays a vital role in the world economy. It has become one of the flourishing industries in the last decade. In 2018, the Travel and Tourism industry has generated around 10.4% of the World GDP, and equally high employment. The tourism section in India was ranked 34^{th} out of 140 countries in the Travel and Tourism Competitiveness Report 2019 [2] which is an improvement in ranking by 6 places over the 2017 report. The price competitiveness of world tourism places India at 13^{th} rank out of 140 countries. This illustrates how the tourism sector in India is the fastest growing service industry with its great potentials for further expansion and diversification. The Travel and Tourism Competitiveness Report 2019 also mentions that India has quite satisfactory air transport infrastructure (ranked 33). According to the calculation by World Travel and Tourism Council, the Indian tourism industry generated 16.91 lakh crore (240 billion US $) or 9.2% of India's GDP in 2018 and supported 42.673 million jobs, (8.1% of its total employment). This pretty much accounts for the reason why the tourism sector in India is an engine for peace-oriented sustainable economic development [15] profusely creating more job opportunities and contributing to a higher standard of living.

Given the scenario, forecasting the foreign tourist arrival(FTA) in India is a significant activity for its beneficiaries and stakeholders' as well as to the researchers. A considerable amount of time series modeling and prediction have been done on the Indian tourism demand. Mishra et al., (2018) [14] and Purna Chandra Padhan (2011) [5] in their study has shown the efficiency of Holt Winters model in forecasting FTA in India. Another study by Smita et al., (2017) [19] has compared two time series forecasting models ARIMA and Holt Winters taking the FTA in millions. In a paper Kumari (2016) [11] , Kriti Kumari has worked with Grey Model and SARIMA to forecast the same. In a very recent study by Kumari et al., (2019) [12] has shown optimal forecasting performances using various alternative time series models.

Indian tourist demand prediction is much of a recent activity, however, forecasting tourism demand has been a very engaging work for researchers for many many years all over the world. A lot of research works, papers and studies have shown very systematic prediction and analysis of the tourist demand comparing various time series models [7] [8]. According to a Tourism literature review by Song et al.,(2008) [18], they have mentioned that depending on the forecasting accuracy, it is evident that no single model consistently outperforms other models in all situations meaning there cannot be any particular deterministic model that works better for the tourism demand all over the world. Surveying various papers on FTA forecasting in India and all over the world we see that models like HW (Holt Winters), ARIMA, SARIMA are more commonly used than Non-linear models like SETAR Claveria et al.,(2014) [8] or Markov-Switching-Regime. Recently several different AI (Artificial Intelligence) methods have been implemented for forecasting the tourist demand in various countries. The most commonly used AI method is ANN (Artificial Neural Networks). ANN has been applied to a large number of time series forecasting problems mostly to capture the non-linearity of the data, but only recently it is being applied on tourism demand forecasting in various countries [16]. Claveria et al., (2014) [8] in their study has shown a very interesting forecast comparison of ARIMA, SETAR, and MLP models for forecasting tourism demand from various countries to Catalonia. The paper also shows how one model outperforms another depending on various time horizons.

Considering the importance of tourism and its influence on economic growth, we have chosen this particular variable for our study. The recent research trends all over the world have inspired us to carry forward a comparative analysis on the foreign tourist arrival in India. For a more justified model comparison, we have taken two different time windows of 6 months and 24 months for our study.

The main objective of this study is to compare the predictive power of two ANN models namely MLP (Multi-layer Perceptron) and LSTM (Long Short Term Memory) to that of a time series Auto Regressive (AR) model. MLP can be broadly classified as a Feed Forward Neural Network and LSTM is considered as a forerunner of Recurrent Neural Networks(RNN). Various researches have shown that both of the ANN models perform generously for time series forecasting. We have used the *Ministry of Tourism* monthly data for Foreign Tourist Arrival in India from January 1988 to December 2019. For the purpose of comparing the prediction from various models, we have used the root mean square error (RMSE), and the mean absolute percentage error (MAPE) for various horizons (6, 24 months) respectively. The result (chapter 3) from our analysis ranks the prediction efficiency for both the time horizons in ascending order of MLP, LSTM and AR models for 24 months forecast.

This report is structured in the following way: Chapter 2 contains the methodological approach, describing both 2.1 time series and 2.2 neural network methods. Chapter 3 results contains the Sub-section 3.1 description of data, followed by 3.2 data analysis and techniques, (both for 3.2.1 time series and 3.2.2 ANN) and 3.3 forecasting performance. Chapter 4 contains the conclusion and 4.1 future research aspects.

Chapter 2

Methodology

2.1 Time Series Modelling

A time series model generally predicts future values based on its own past and a random disturbance term. Modeling a univariate seasonal time series accurately is a challenging task. In this work, we have followed a stepwise approach to select the correct model for our variable.

2.1.1 Step 1 : Obtaining a stationary time series

For any time series, the main objective is to first convert the series into a stationary time series. The process of testing and converting into a stationary series is initiated after removing any seasonality from the data. To that end, one uses seasonal dummy variable regression to check for seasonal peaks in the data. The equation for seasonal dummy regression is:

$$y_t = \beta_0 + \beta_1 t + D_{1t} + \beta_2 D_{2t} + ... + \beta_{12} D_{12t} + u_t, u_t \sim N(0, \sigma^2) \tag{2.1}$$

where D_{it} are the dummy variables for each month defined by

$$D_{it} = \begin{cases} 1 & \text{if } t_{th} \text{ observation} \rightarrow i_{th} \text{ month} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

If the coefficient of one or more dummy variables turns out to be significant at α level of significance, seasonality in the data is confirmed. The seasonal peaks are removed using various filters like **classical decomposition** for additive time series $y_t = T_t + S_t + u_t$ or **Ratio to moving average** filter for multiplicative time series $y_t = T_t \times S_t \times u_t$. Once the seasonal peaks are demolished, the presence of any stochastic trend is determined using a **Augmented Dickey Fuller test** (Dickey, Fuller 1979) [20]. For a trend stationary process, the ADF test equation is given by:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \cdots + e_t,\ e_t \sim WN(0, \sigma^2) \tag{2.3}$$

The null hypothesis for the ADF test is in favour of the presence of a unit root ($\gamma = 0$), while the alternate hypothesis says that there is no unit root in the data i.e. ($\gamma < 0$). The null and the alternate hypothesis are respectively:

$$H_0 : \gamma = 0 \tag{2.4}$$

$$H_1 : \gamma < 0 \tag{2.5}$$

The test statistic is checked at a chosen significance level say α. The values of the test statistic at various levels of significance is given in the Table 2.1.

Table 2.1: ADF test statistic significance levels

	1%	5%	10%
tau 3	-3.98	-3.42	-3.13
phi 2	6.15	4.71	4.05
phi 3	8.34	6.30	5.36

If a unit root is detected in the deseasoned series, it is thereafter removed by differencing. The process continues till we obtain a series free from the stochastic trend. Once the goal is accomplished, we estimate the deterministic trend in the data by applying a **curve fitting** method. This trend is subsequently removed from the data rendering a stationary time series.

2.1.2 Step 2: Model selection

Generally for a comparative analysis, model selection is done on the in sample. The correctly specified model is obtained by the Box, Jenkins (1971) [4] procedure. It primarily does a correlogram analysis, testing the significance of the autocorrelation and the partial autocorrelation of the lags followed by residual analysis. The various time series models are listed below:

Auto Regressive (AR(p)) model:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t \tag{2.6}$$

where $\phi_0, \phi_1, \ldots, \phi_p$ are constants and ϵ_t is white noise.

Moving Average (MA(q)) model:

$$y_t = \theta_0 + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \tag{2.7}$$

where $\theta_0, \theta_1, \ldots, \theta_q$ are contants, and ϵ_t is noise.

Auto regressive moving average (ARMA(p,q)) model:

$$y_t = c + \theta_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{2.8}$$

where $\theta_1, \ldots, \theta_q, \phi_1, \ldots \phi_p\ and\ c$ are constants.

Correlogram analysis:

In correlogram analysis, the model selection is based on analysing the autocorrelation and partial autocorrelation of the various lags of our time series. So before describing the process we head start with defining the acf and pacf.

The **autocorrelation**(ACF) between any two lags, x_t and x_{t-k} is defined by $\rho_k = \frac{\gamma_k}{\gamma_0}$ where

$$\gamma_k = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{N} \quad \text{and} \tag{2.9}$$

$$\bar{x} = \frac{\sum_{t=1}^{N} x_t}{N} \tag{2.10}$$

Similarly the **partial autocorrelation**(PACF) between any two lags x_t and x_{t-k} is defined as the autocorrelation between x_t and x_{t-k}, conditioned on $x_{t-k-1}, \ldots, x_{t-1}$, the set of observations that come between the time points t and t-k.

It is explicitly given by

$$\phi_{11} = \hat{\rho}_1 \tag{2.11}$$

$$\hat{\phi}_{kk} = \frac{\hat{\rho}_k - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}_{k-j}}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{k-1,j}\hat{\rho}_{k-j}} \tag{2.12}$$

where $\hat{\phi}_{kj} = \hat{\phi}_{k-1,j} - \hat{\phi}_{kk}\hat{\phi}_{k-1,j-1}, \quad j = 1, 2, \ldots, k-1.$

By the Box, Jenkins (1971) [4] procedure of model selection, a stationary time series is modelled as a $AR(p)$ process, if the acf of the lags are diminishing, but the $pacf$ cuts off at lag $\leq p+1$, similarly as an $MA(q)$ process can be identified if acf cuts off after lag $\leq q+1$ onwards whereas the $pacf$ is diminishing. For an $ARMA(p,q)$ process both the acf and the $pacf$ is diminishing. IF the model happens to be an $ARMA(p,q)$ model, the lags p and q are then selected depending on the minimum Akaike Information Criterion(AIC), or Bayesian Information Criterion(BIC). Note that here, *cuts off after a particular lag means that the acf, pacf values become insignificant after a particular lag.* The following table summarizes the model selection procedure.

Table 2.2: ACF and PACF model response

	ACF	PACF
AR(p)	diminishing	cuts off after (p+1) lag
MA(q)	cuts off after (q+1) lag	diminishing
ARMA(p,q)	diminishing	diminishing

The significance of the acfs $(\rho_1, \rho_2, \ldots)$ at various lag is tested by **Ljung Box test**, whose null and alternate hypothesis are respectively:

$$H_0 : \rho_1 = \rho_2 = \cdots = \rho_k = 0, \tag{2.13}$$

$$H_1 : \rho_i \neq 0 \;\; \forall\, i = 1, 2, ...k, \tag{2.14}$$

The Ljung-box test statistic follows a χ^2_k distribution given by :

$$Q(k) = n(n+2)\left(\sum_{j=1}^{k} \frac{\hat{\rho_j}^2}{n-j}\right) \sim \chi^2_k \tag{2.15}$$

Similarly, the significance of the $pacf$s $(\phi_1, \phi_2, \dots)$ at subsequent lags at α level of significance is tested by the following null and alternate hypothesis:

$$H_0 : \phi_1 = \phi_2 = ... = \phi_k = 0, \tag{2.16}$$

$$H_1 : \phi_i \neq 0 \ \forall i = 1, 2, ...k \tag{2.17}$$

The test statistic of the $pacf$ follows a Standard Normal distribution given by:

$$Q(k) = \frac{\hat{\phi}_k}{SE(\hat{\phi}_k)} \sim N(0, 1) \tag{2.18}$$

Redisual Analysis:

After intuitively selecting the correct model by a correlogram analysis, the autocorrelation coefficients of the residues are again tested using Ljung Box test [4]. The test statistic now follows a χ^2_d distribution, at $d = k - p - q$ degrees of freedom. No significant autocorrelation between the lags in the residues would suggest that the model is correctly specified.

2.1.3 Step 3: Forecast evaluation matrices

For a comparative analysis, the performance of the forecasting model is tested on the hold out sample and is evaluated through various forecast accuracy matrices. Forecast accuracy is the measure of the difference between the actual value and the predicted value at time t. It is often referred to as residues from forecast given by $e_t = Y_t - \hat{Y}_t$ where Y_t and $\hat{Y}_t$ are respectively the actual value, and the predicted value. Two commonly used forecast evaluation matrices are RMSE and MAPE. RMSE is the primary measure to evaluate the error while forecasting. However, MAPE is also often used to compare between models as it is independent of scale.

$$\textbf{RMSE} : \sqrt{\frac{\sum_{i=1}^{k}(Y_i - \hat{Y}_i)^2}{n}} \tag{2.19}$$

$$\textbf{MAPE} : \frac{1}{n}\sum_{i=1}^{k}\frac{|(Y_i - \hat{Y}_i)|}{|Y_i|} \times 100\% \tag{2.20}$$

2.2 Artificial Neural Networks

The first brick in the foundation of an advanced future of Artificial Neural Networks (ANNs) was laid by Warren McCulloch and Walter Pitts, 1943 [3]. The idea of ANNs is to build a computational model vaguely based on the structure and functioning of biological neural networks that constitute animal brains. The main purpose of this is to replicate the way humans analyze and work. The basic structure of an ANN consists of nodes known as artificial neurons that are grouped into three main layers, namely, an input layer, one or more hidden layers or dense layers and an output layer. All the neurons in each layer are connected to every other neurons in the next layer and thus we say that the network is fully connected.

A given neuron in the hidden layer takes the weighted sum of its inputs and a bias term

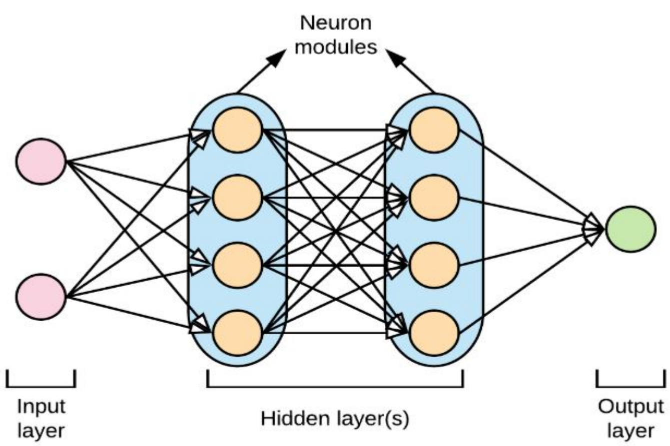

Figure 2.1: Structure of an ANN with two hidden layers.

and passes it through a non-linear activation function. This output of one neuron becomes the input of other neurons in the next layer. This process continues until we get the outputs of the neurons of the last hidden layer. Finally, by applying a suitable application function to the output layer, the outcome of the model will be predicted. This outcome of the model is compared to the original result and the process is repeated several times to enhance the accuracy of the output.

The main feature of a time series data is that the value of the present observation is dependent on the values of the past. The use of an ANN for forecasting time series implies that the input nodes are connected to several past observed values which are supposed to be sufficient enough to predict the values at future time steps. There are different types of artificial neural networks based on their structures and functions. In this paper, we will analyze the prediction accuracy of two ANN models namely, MLP and LSTM based on our data.

2.2.1 Multilayer Perceptrons (MLP)

MLP is a type of feedforward neural network, wherein connections between the nodes do not form a cycle i.e. the information flows only in one direction from the input layer to the output layer. It was one of the first and simplest types of ANN devised. It has the same basic structure of an ANN with one or multiple hidden layers consisting of neurons as in figure 2.1[1]. The structure of each neuron consists of five components: inputs, weights, sum, activation function and the output [17].

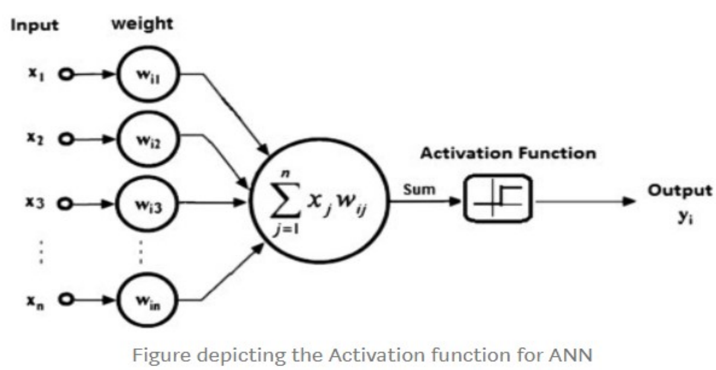

Figure 2.2: Structure of an artificial neuron.

The output of the neuron depends on the weights of the inputs and the activation function of the neuron as shown in figure 2.2 [2]. The output y_i of the ith neuron with n inputs is illustrated mathematically by

[1]https://ekababisong.org/gcp-ml-seminar/deep-learning/

[2]https://becominghuman.ai/artificial-neuron-networks-basicsintroduction-to-neural-networks-3082f1dcca8c.

$$y_i = \sigma(\sum_{j=0}^{n} x_j w_{ij})$$

where σ is the activation function and the bias is represented by $j = 0$.
The activation function is used to scale the output to be within a certain range. Sigmoid function (0 to 1), hyperbolic tangent function (-1 to 1) and rectifier (0 to $+\infty$) are some of the commonly used activation functions.
In the context of forecasting, the model can establish the temporal relationships within the data. More the number of neuron and hidden layers, more the model will be able to learn complex tasks, but too many will result in overfitting of the model. It is therefore important to optimize the number of neurons and hidden layers.

Model training and forecasting

The learning process of the model is an optimization exercise which is to minimize a specified loss function, which is generally the mean absolute error (MAE) or the mean squared error (MSE) [17], by tuning the weight parameters of the network. The optimization is done using an algorithm called the gradient descent [17] in which the gradient of the loss function is calculated with respect to the weights and bias of the neurons. The weights are updated in the opposite direction of the error gradient to minimize the error. A technique called backpropagation is used to compute the gradients which enables it to see how the error from the loss function is propagated backward to each preceding neuron layer. If L denotes the loss function and γ represent the learning rate of the model, then

$$\theta' = \theta - \gamma \frac{\delta L(\theta)}{\delta \theta}$$

where θ' and θ represent the updated and previous weights. This process is repeated several times (called epochs) over the training data to optimize the weights and minimize the error. MLP can conduct both univariate and multivariate forecasts by customizing the number of nodes in the output layer.

2.2.2 Long Short Term Memory Networks (LSTMs)

LSTM is a type of Recurrent Neural Network (RNN) which is a descendant of MLP. The structure of RNN was first modeled by Elman, 1943 [9]. It is called recurrent because there exists a backward connection between the output of the hidden layer and its inputs, that is, it has memory property. This forecasting model utilizes the memory of past forecast values to make future forecasts. The recurrent loops in an RNN are shown in figure 2.3, where the decision made in the current time step is affected by the output of the previous time step.

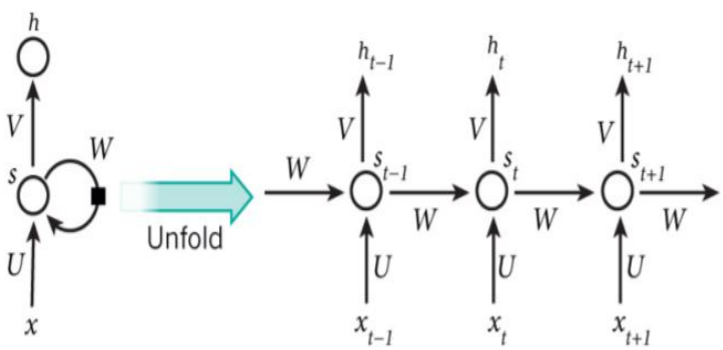

Figure 2.3: Recurrent Neural Network [17]

x and h denote the input and output data respectively and the internal state is denoted by s. The weight matrices between the input and hidden layers, the hidden and output layers, and hidden layers are denoted by U, V and W respectively. It can be observed that the current output depends on all the previous inputs, which enables RNNs to incorporate temporal dependencies in the data. The structure of the RNN can be mathematically shown by the two equations below.

$$s_t = \sigma(Ux_t + Ws_{t-1} + b_s)$$
$$h_t = \sigma(Vs_t + b_h)$$

where σ is the activation function and b denotes the neuron bias.

Model training of RNN

Similar to MLP, RNN also uses backpropagation and gradient descent to update the weight parameters to optimize the loss function. However, in the case of RNN there is a need for backpropagating the error through time. But, in practice, it is found that RNNs perform very poorly even if the inputs and output are separated by only 10 time steps [17].

LSTMs, introduced by Hochreitzer Schmidhuber [9], 1997, are capable of learning long term dependencies. That is, they can remember information for long periods. LSTM is a gated architecture that solves the vanishing gradient problem of RNN by using gated RNNs. The access to the memory state of each LSTM unit is restricted by gates which are of three types, input gate, forget gate and output gate. The structure of an LSTM unit is shown in figure 2.4, where f, i and o denotes the forget, input and output gate functions respectively and c denotes the memory state of the cell.

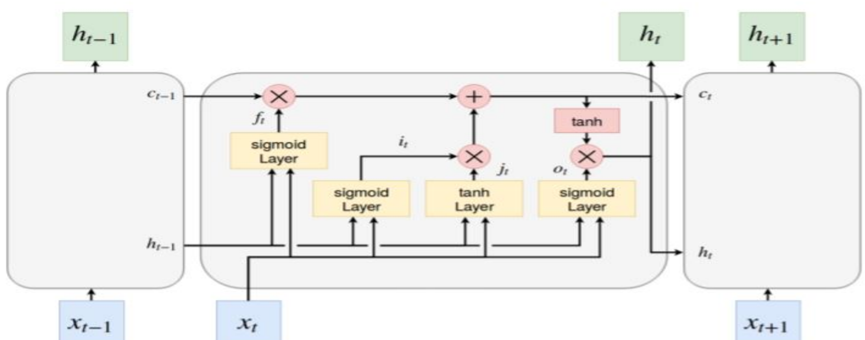

Figure 2.4: LSTM unit with gated Connections at a single time step t [10]

The gate functions have a value between 0 and 1, which helps the gates to manipulate the memory state of the cell. For example, a forget gate function value of 1 will keep the memory state intact but a value of 0 will cause the cell memory state to be totally forgotten.

The following mathematical equations show the various functions in a LSTM unit.

$$
\begin{aligned}
j_t &= \tanh W_j x_t + R_j h_{t-1} + b_j && \text{(input)} \\
i_t &= \sigma(W_i x_t + R_i h_{t-1} + b_i) && \text{(input gate)} \\
f_t &= \sigma(W_f x_t + R_f h_{t-1} + b_f) && \text{(forget gate)} \\
o_t &= \sigma(W_o x_t + R_o h_{t-1} + b_o) && \text{(output gate)} \\
c_t &= j_t \odot i_t + c_{t-1} \odot f_t && \text{(cell state)} \\
h_t &= \tanh(c_t) \odot o_t && \text{(output)}
\end{aligned}
$$

Where W and R denote the input and recurrent weight matrices for an LSTM unit and sigma represents the activation function. Here, the operator $\odot$ denotes element wise multiplication. Thus, LSTM is able to learn long term temporal dependencies in the data efficiently by having fine control over the cell's memory state.
LSTMs can also be used for both univariate and multivariate forecasting as both input and output are in the form of vectors.

Chapter 3

Results

3.1 Description of Data

The data used in this study is a monthly data of tourist arrival from foreign countries to India over the time period from January 1988 to December 2019 provided by the Ministry of Tourism. Our data has altogether 384 observations of 342 months spanning over 32 years. The plot is as shown in figure 3.1. The data is collected from CEIC [1], which has been a trusted partner to help navigate the world of various types of data for over two decades.

The entire data analysis, visualization and modeling are done in R-4.0.2 in the RStudio integrated development environment (IDE), which is one of the best user-friendly environments to do exploratory data analysis. R is developed by academics and scientists and is the right tool for data science because of its powerful communication libraries.

3.2 Data Analysis

3.2.1 Time Series Analysis

The data visualization in 3.1 in section 3.1, gives us an intuitive understanding of the presence of seasonality. The zoomed in plot two of figure 3.1 prominently shows the monthly seasonal peaks, generally in the months of December, January. The plot also shows an increasing variance, which affirms the multiplicative property of the series.

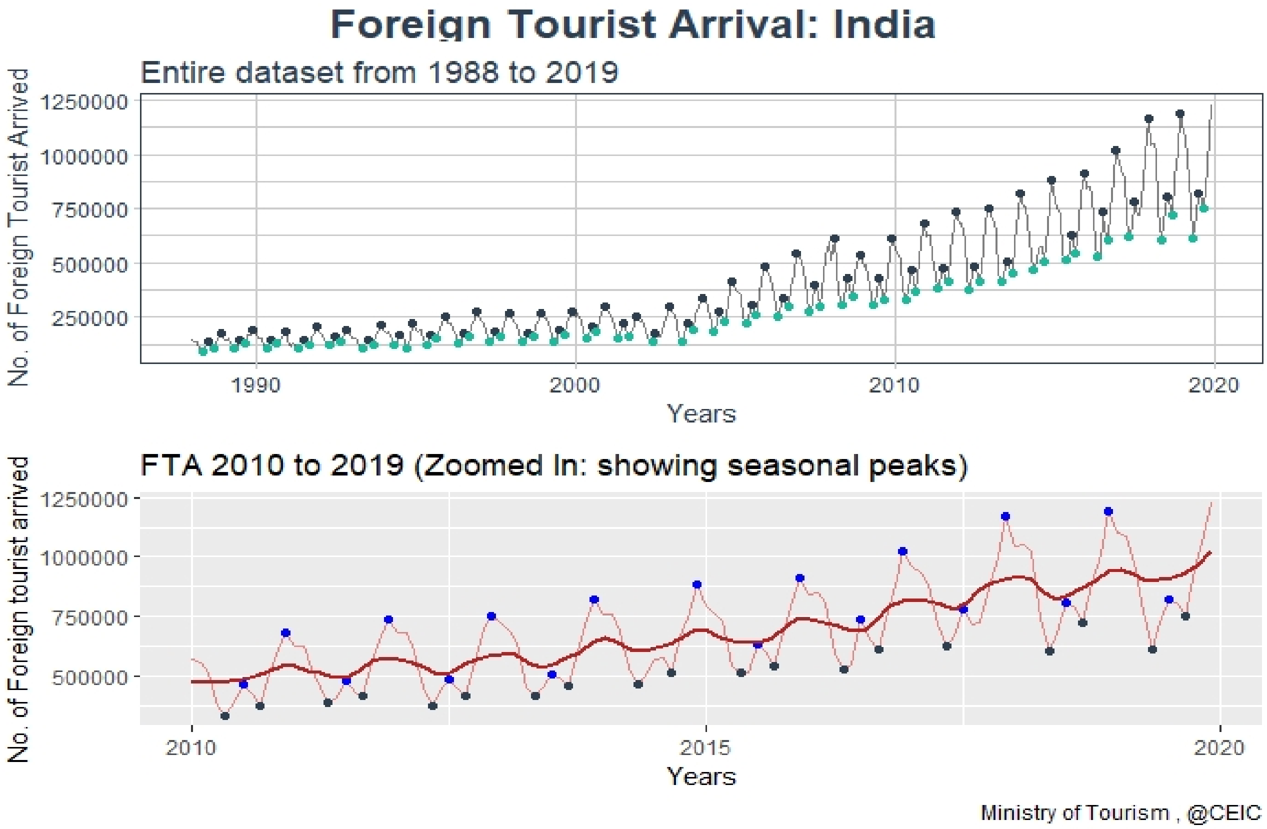

Figure 3.1: Monthly foreign tourist arrival in India

Firstly, we have converted the data into a additive time series by taking log transformation, next we have confirmed the presence of seasonality in the data by applying dummy variable regression equation (2.1), (2.2). The seasonality is then removed by calculating the seasonal indices $S_1, S_2, \ldots, S_{12}$ using the Ratio to MA filter. These seasonal indices are then removed from the data. On the deseasoned data we have applied the Augmented Dickey Fuller(ADF)[20] test to check the presence of a Stochastic trend(unit root).

Table 3.1: Results from ADF test

Variable	ADF test statistic
y_t (deseasoned)	-2.08
$y_t^{'}$ (diff)	-9.63^{***}

The value of the test statistic obtained is $-2.08 > -3.98$ (Table 2.1 [1]). This shows that we fail to reject the null hypothesis (equation 2.3) at 5% level of significance, concluding the

[1] *** :10 %, ** : 5% and *** : 10 % level of significance**

presence of a unit root. The stochastic trend is then removed by taking difference i.e. $y_t' = y_t - y_{t-1}$ This differenced series is again tested using ADF test. This time we obtain a test statistic value $-9.63 < -3.98$, thus confirming that the stochastic trend is now completely removed from the data. Next, we have fitted a linear trend $y_t = a + bt + u_t$ to the data and tested the null hypothesis $H_0 : b = 0$ using linear regression. The result suggests that at $\alpha = 0.05$ we can reject the null hypothesis, showing that there is no deterministic trend in the data. The data thus obtained is stationary.

Next, as mentioned in section 2.1.2 we divide this stationary time series into the in sample and hold out sample. For our work, we have taken two different prediction horizons of 24 months and 6 months respectively. For 24 months prediction, the **in sample** is taken from 01-1988 to 12-2017 (30 years), and the **hold out sample** is taken from 01-2018 to 12-2019 (2 years), and for 6 months prediction, the in sample is taken from 01-1988 to 06-2019 (31.5 years) and the hold out sample (0.5 years) is taken from 07-2019 to 12-2019. The model for the separate in sample turns out to be the same, so we shall describe the modeling technique for 6 months prediction.

On the in sample for 6 months prediction, as mentioned in section 2.1.2 we perform the correlogram analysis. ACF test statistic column in table 3.2 shows that all the test statistic values are significant at $\alpha = 0.05$ level implying that the null hypothesis (equation 2.13) for all the lags are rejected i.e. the acf is diminishing. On the other hand, the PACF test statistic column in the table 3.2 shows that null hypothesis (equation 2.16) is alternately rejected and accepted (alternately significant), but after lag 10, all the null hypothesis are accepted at $\alpha = 0.05$ level of significance. According to the correlogram analysis methodology in section 2.1.2, table 2.2, this implies that we can model our time series as a AR(10) model. After fitting a AR(10) model to the insample, the acfs of the residues are calculated. Note that for the residual table 3.3 the lag values are obtained from lag 12, as because the first 10 lag are "NA" due to fitting a AR(10), and the 11th lag is 1. model. The residual table is as shown in Table 3.3 clearly shows that all the lags are insignificant, i.e. we fail to reject the null hypothesis for all k lags, (equation 2.13) at $\alpha = 0.01$ level of significance, which means that there is no significant autocorrelation between the residual lags. This implies that our model is correctly specified.

Table 3.2: ACF and PACF significance.[a]

Lag	ACF Test Statistic	PACF Test Statistic
1	6.452565**	2.5301165**
2	19.633253**	4.0090804**
3	20.48812**	2.1343201**
4	22.839477**	0.2901904
5	25.454775**	2.0256298**
6	35.803336**	3.8180161**
7	36.136912**	2.69099**
8	36.774303**	1.8581609
9	37.54307**	0.654882
10	40.070703**	2.3814725**
11	41.19073**	0.5527202
12	48.755986**	1.2918263
13	49.789723**	1.269126
14	50.099938**	1.1055511
15	51.561362**	0.2441403
16	52.615935**	1.044147
17	52.900202**	0.3722389
18	55.70178**	0.2608528

[a] * :1 %, ** : 5% , *** : 10 % significance level

Table 3.3: The residual table.[a]

Lag	Residual ACF	P values	Test Statistic
12	0.02755	0.024614	5.050872
13	0.084497	0.069394	5.33592
14	0.026913	0.148767	5.336261
15	-0.00093	0.158209	6.606388
16	-0.05666	0.236626	6.791345
17	0.02159	0.334461	6.855048
18	-0.01265	0.30838	8.282063
19	-0.0598	0.26958	9.935533
20	-0.06428	0.192384	12.386553
21	-0.07816	0.250054	12.547976
22	-0.02003	0.112141	16.858197
23	0.103356	0.116485	17.973911
24	0.052511	0.123936	18.973236
25	0.049627	0.082457	21.816821
26	0.083595	0.02486	27.507974
27	-0.1181	0.036126	27.512549
28	-0.00334	0.042571	28.207539

[a] * :1 %, ** : 5% , *** : 10 % significance level

Therefore, the correctly specified time series **Auto Regressive(10)** model for Foreign Tourist Arrival is:

$$\begin{aligned}\mathbf{y_t} =& -\mathbf{0.2y_{t-1}} - \mathbf{0.24y_{t-2}} - \mathbf{0.14y_{t-3}} - \mathbf{0.05y_{t-4}} - \mathbf{0.16y_{t-5}} - \mathbf{0.21y_{t-6}} \\ & \mathbf{0.13y_{t-7}} - \mathbf{0.08y_{t-8}} - \mathbf{0.01y_{t-9}} - \mathbf{0.08y_{t-10}} + \hat{\epsilon}_\mathbf{t},\end{aligned} \tag{3.1}$$

Next, we forecast the value for 24 months and 6 months and find the forecasting error. This will be discussed in details in the result section 3.3.

3.2.2 Artificial Neural Networks Modelling

In this section, we will model two types of ANNs namely, MLP and LSTM based on our FTA data. First we will formulate a MLP model for our data and then keeping the same network structure we will formulate our LSTM model. Here we will work with the stationary data that we obtained in section 2.1.1. We divided the data into two parts: the training set (with which we trained our model) and test set (to compare the predicted values and calculate the value of the loss function). The nodes in the input layer are lagged values of our time series data that will help the model to predict future values. For this, we checked how significantly the lagged values at time points $t-k$ will affect the observation at time t, where k=1,2,. . . ,12.

There is no hard and fast rule for deciding the number of hidden layers and the number of neurons for each layer in an artificial neural network. There always exists an opportunity to tune our model further for better prediction. However, it is always recommended to select the network which performs best with the least possible number of hidden neurons. Therefore, for MLP, to check the required number of neurons for our hidden layer we formulate our model using 1 to 15 neurons and collect the MSE values. This process is repeated for 50 epochs for each number of neurons and the mean MSE is calculated for them all. We choose the number of neurons that give the least MSE.

Case 1: For 24 months prediction.

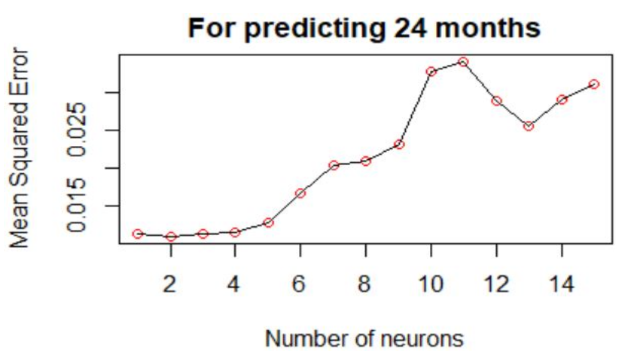

Figure 3.2: Plot of MSE corresponding to the number of neurons

Our training set here consists of 360 data points (from 01-1988 to 12-2017) and our test set consists of 24 data points (from 01-2018 to 12-2019 On checking the significance of lagged observations in our training set,we found that for k=1,2,3,4,5,6,7,8,10, the observation values at $t-k$ was affecting the value at t. Therefore, our input layer consists of nine nodes. And the form of the training set is like $\{(x_{t-10}, x_{t-8}, x_{t-7}, ..., x_{t-1}) \rightarrow x_t\}$

The MSE values corresponding to the number of neurons is obtained and the plot is shown in figure 3.2. The least MSE value comes for 2 neurons and hence our hidden layer consists of two neurons. The structure of our model is shown in figure 3.3. The MLP model is trained based on the training set and predictions are made for 24 months. The predicted values are compared with the actual values of the test set and the RMSE and MAPE (section 2.1.3) values are noted.

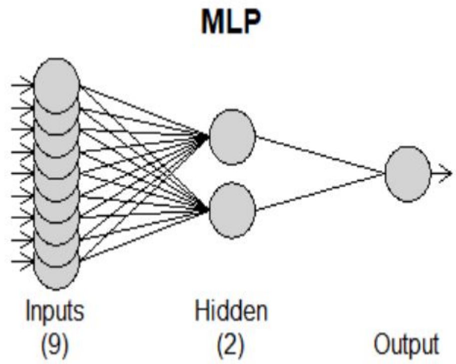

Figure 3.3: Structure of MLE model for predicting 24 months

Case 2: For 6 months prediction.
Our training set here consists of 378 data points (from 01-1988 to 06-2019) and our test set consists of 6 data points (from 07-2019 to 12-2019).

We found that for k=1,2,3,4,5,6,7,8,10,12 the observation values at $t-k$ was affecting the

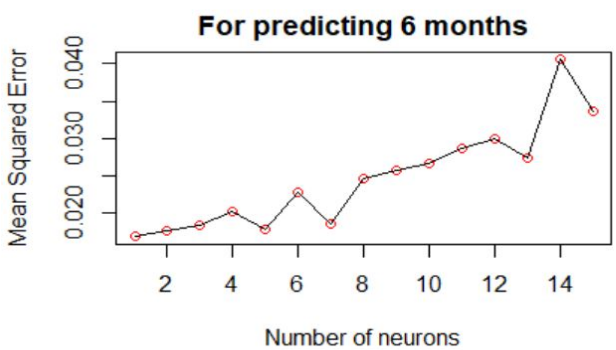

Figure 3.4: Plot of MSE corresponding to the number of neurons

value at t. Therefore, our input layer here consists of ten nodes.And the form of the training set is like $\{(x_{t-10}, x_{t-8}, x_{t-7}, ..., x_{t-1}) \rightarrow x_t\}$

The MSE values corresponding to the number of neurons is obtained and the plot is shown

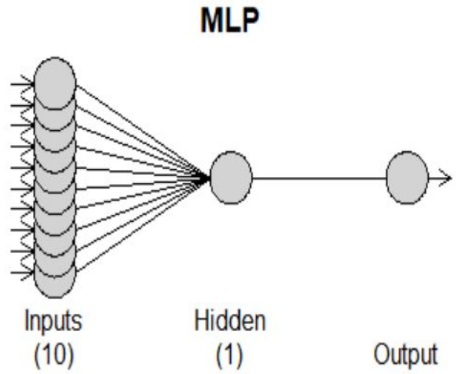

Figure 3.5: Structure of MLE model for predicting 6 months

in figure 3.4. Here, the least MSE value comes for 1 neuron and hence our hidden layer consists of one neuron. The structure of our model is shown in figure 3.5. The model is trained based on the training set and predictions are made for 6 months. The predicted values are compared with the actual values of the test set and the RMSE and MAPE values are noted.

Now, for LSTM we reproduce the same network structure as MLP for both 24 months and 6 months respectively. That is, for 24 months our LSTM model has an input layer consisting of nine nodes, one hidden layer consisting of two neurons and an output layer with one neuron. Similarly, for 6 months our LSTM model has an input layer consisting of ten nodes, one hidden layer consisting of one neuron and an output layer with one neuron. But as we see in section 2.2, the inputs in an LSTM model are outputs of a *tanh* function. So, we need to first normalize the observations such that they range between -1 to +1. After that we model our normalized training data for 24 months and 6 months on their respective LSTM models. Predictions are made for 24 months and 6 months and are compared with their respective validation sets. The RMSE and MAPE values are then calculated for both the models' predictions.

3.3 Forecasting Performances.

The forecasting performance of the time series Autoregressive model is evaluated in this section relative to the two Artificial Neural Network models MLP and LSTM. For an unbiased comparison between all the three models, we have used the stationary time series (section 2.1.1) for modelling all three. First, we have modelled the time series as described in section 3.2.1 which turned out to be a AR(10) (refer 3.1) model. Next as mentioned in section 3.2.2, in plots 3.5 and 3.2 we have decided upon the structures our MLP model (9:2:1 and 9:1:1 for 24 months and 6 months prediction respectively) by choosing the minimum number of neurons through lowest MSE criterion. To maintain a parity in models we have used similar architecture in LSTM (section 3.2.2). Their performances are judged based on RMSE and MAPE both for the 6 months and 24 months prediction. Figure 3.6 shows the actual values and model forecasted values for all the three models for 6 months prediction. The plots of the predictions of different models along with the original time series (named as FTA) are shown in plot 3.8 for 24 and 6 months respectively.

6 Month Prediction (2019)	Actual	LSTM	MLP	AR(10)
July	817455	851326.7	885013.1	894251.1
August	798587	748402.7	820577.3	823918.7
September	750514	725638.9	746769.1	752514.4
October	944233	947581.5	948888.4	963521.6
November	1091946	1110533.0	1105662.7	1132446.0
December	1225672	1234828.3	1245140.4	1268279.2

Figure 3.6: 6 month prediction value for LSTM, MLP and AR model

It is evident from the predicted numbers in table 3.6 and the plots 3.8, that all the three models have performed well. The RMSE and MAPE values in Table 3.7 indicate that for both the time horizons 6 months and 24 months, the ANN models namely LSTM and MLP have outperformed the naive Auto Regressive time series model. For an intra-comparison between ANN models we see that 6 months forecasting, LSTM has performed marginally better than MLP with respect to the RMSE value. However, for 24 months prediction MLP model performed the best among the three with rest to both the forecasting matrices.

	RMSE		MAPE	
	6 months	*24 months*	*6 months*	*24 months*
AR(10)	41575.27	51369.94	3.68	4.89
MLP	30688.03	31962.67	2.47	2.81
LSTM	28062.57	39484.9	2.75	3.89

Figure 3.7: RMSE and MAPE values for different models.

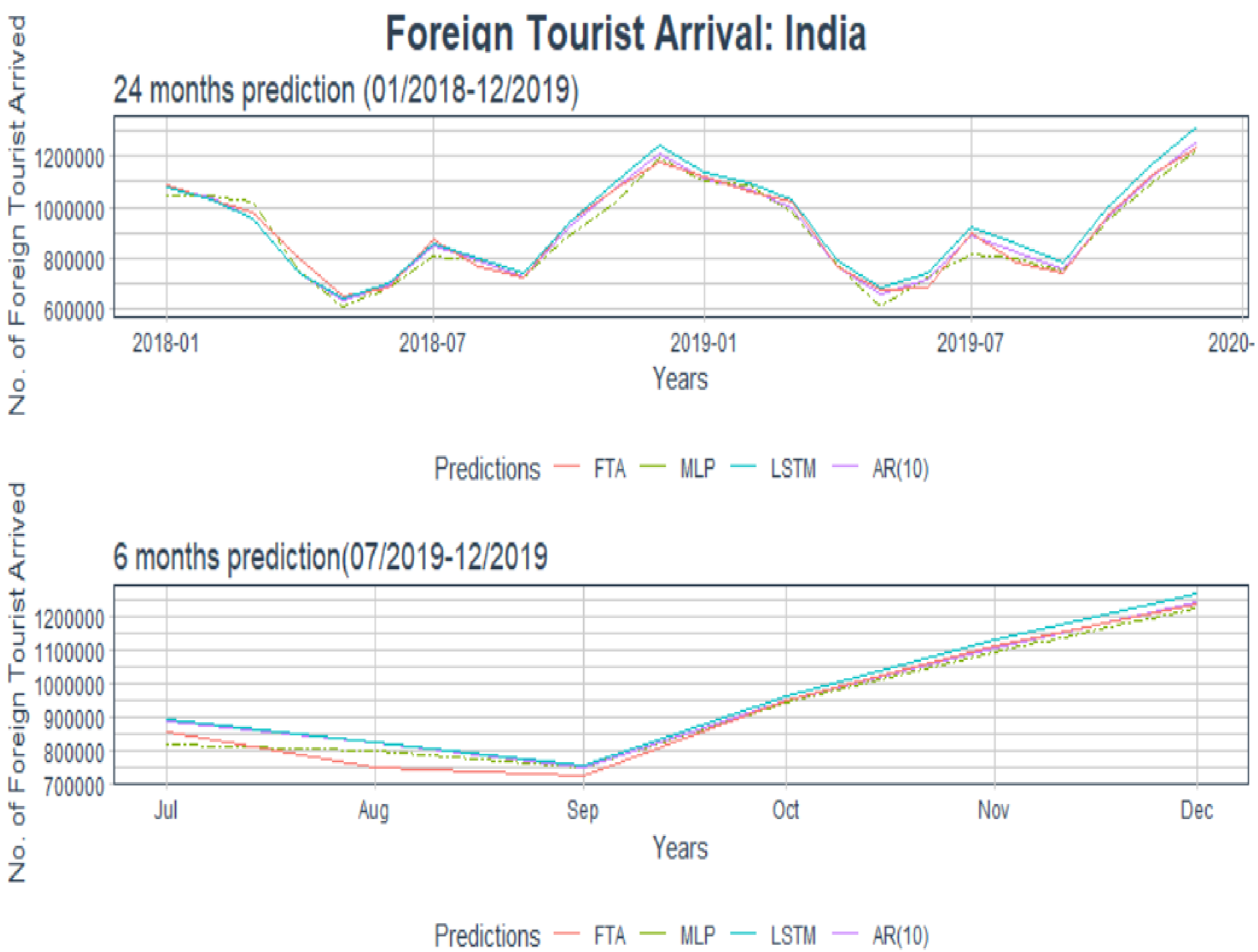

Figure 3.8: Plot comparing the predicted results with actual value.

Chapter 4

Conclusions

The tourism sector in Indian is flourishing at an enormous rate. The rising social and economic significance of the tourism sector in India and the modern forecasting trends have instigated in us the zeal to work on this particular project of evaluating the forecasting performance of neural networks (MLP and LSTM) relative to time series model (AR). We have used two different forecasting window of 6 months and 24 months to find out how well behaved our model performances are for short term and long term prediction. It is a prevalent belief that nonlinear models like ANN models outperform the linear models for economic modeling. The result we obtained in our study (section 3.3) aligns with that belief. Many researchers like Cho (2000) [6], and Law (2000) [13] have also shown similar results for tourism demand forecasting. Nevertheless, there also exist researches on tourism demand of various other countries like in Claveria et al., (2014) [8] which has shown better performance of a time series model over a ANN model. However considering the suggestion by Song et al,.(2008) [18], that no model consistently outperforms another model depending on model accuracy, we see that even for our ANN models, LSTM outperformed MLP for 6 months prediction concerning RMSE value under similar circumstances.

The main aim of time series modeling is to capture the persistence behavior of the stationary series. Since we have used the stationary series for all the three models, we believe that the particular outcome that we obtained is due to the capacity of the Neural Network models to capture the noise or the persistence behavior better than time series model which are linear in parameter. This can be owing to the fact that the stationary series we obtained, still possessed a certain level of unknown nonlinear dependencies. However, since the time series auto-regressive model could only capture the linear dependencies in the data, it could not perform as efficiently. Also there can be lags of unknown duration between two important events of a time series data. LSTM, because of its memory property, is relatively insensitive to such gap length in comparison to the other models, hence having the ability to give a better prediction if properly tuned. Considering all these facts, on a concluding note of this small project, we can therefore say that for modelling the foreign tourist arrival in India, ANN models (MLP LSTM) perform better than a linear benchmark Auto Regressive time series model under stationary conditions.

4.1 Future Scope

It should be taken into account that the ANN models we used here can be engineered, i.e. they can be hypertuned and the model performance can be improved. Also there are many prevailing ANN architectures for economic modeling that can be implemented later on. One can also model the data using various other linear and non linear time series models like Holt Winters and Self-Exciting Threshold Auto Regressive (SETAR) model to improve the time series forecasting performance. One can also take into consideration various factors affecting tourist arrival in India, such as weather indices like average temperature, rainfall etc, and can perform multivariate analysis. We consider all these ideas as a implementable future scope of this project.

Bibliography

[1] Global Economic Data, Indicators, Charts & Forecasts — CEIC.

[2] The Travel & Tourism Competitiveness Report 2019 Travel and Tourism at a Tipping Point. Technical report.

[3] Michael A. Arbib. Warren McCulloch's search for the logic of the nervous system. *Perspectives in Biology and Medicine*, 43(2):193–216, 2000.

[4] GEP Box, GM Jenkins, and GC Reinsel. Time series analysis: forecasting and control. 2011.

[5] Purna Chandra Padhan. Forecasting International Tourists Footfalls in India: An Assortment of Competing Models. *International Journal of Business and Management*, 6(5), 2011.

[6] Vincent Cho. A comparison of three different approaches to tourist arrival forecasting. *Tourism Management*, 24(3):323–330, 2003.

[7] Fong-lin Chu. Forecasting tourism demand with ARMA-based methods. *Tourism Management*, 30(5):740–751, 2009.

[8] Oscar Claveria and Salvador Torra. Forecasting tourism demand to Catalonia: Neural networks vs. time series models. *Economic Modelling*, 36:220–228, 2014.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] B Ravi Kiran, Dilip Mathew Thomas, Ranjith Parakkal, Projeto De Gradua, Max Ferguson, Yung-tsun Tina Lee, Kincho H Law, Shiyang Zhou, Youping Chen, Dailin Zhang, Jingming Xie, Yunfei Zhou, Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, Nassir Navab, Shahrzad Faghih-Roohi, Siamak Hajizadeh, Alfredo Nunez, Robert Babuska,

Bart De Schutter, Jan Paul Assendorp, Yohei Takada, Tokiko Shiina, Hiroyasu Usami, Yuji Iwahori, Ehab A. Kholief, Samy H. Darwish, M. Nashat Fors, Liu Ri-Xian, Yao Ming-Hai, Wang Xian-Bao, Jonathan Masci, Ueli Meier, Dan Ciresan, Jürgen Schmidhuber, Gabriel Fricout, Lili Liu, Rui Jun Yan, Varun Maruvanchery, Erdal Kayacan, I. Ming Chen, Lee Kong Tiong, Hua Yang, Shuang Mei, Kaiyou Song, Bo Tao, and Zhouping Yin. Deep learning for anomaly detection in multivariate time series data. *The Ninth International Conferences on Pervasive Patterns and Applications Defect 2017*, 13(1):1–12, 2017.

[11] K. Kriti. Forecasting foreign tourist arrivals in India using different time series models. *International Journal of Business and Management*, 15, 38-43(August), 2016.

[12] Kriti Kumari, Shalini Chandra, and Samarjit Kar. Forecasting foreign tourist arrivals to India using alternative forecasting combinations. 4(5):20–24, 2019.

[13] Rob Law. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *Tourism Management*, 21(4):331–340, 2000.

[14] P K Mishra, Himanshu B Rout, and B B Pradhan. Seasonality in Tourism and Forecasting Foreign Tourist Arrivals in India. *Iranian Journal of Management Studies*, 11(4):629–658, 2018.

[15] P. K. Mishra and J. K. Verma. Tourism and peace in economic development perspective of India. *Journal of Environmental Management and Tourism*, 8(4):927–934, 6 2017.

[16] Lindsay W. Turner Sen Cheong Kon. Neural network forecasting of. 11(3):301–328, 2005.

[17] Akash Singh. Anomaly Detection for Temporal Data using Long Short-Term Memory (LSTM). *STOCKHOLM, SWEDEN: Information and Communication Technology*, 2017.

[18] Haiyan Song and Gang Li. Tourism demand modelling and forecasting-A review of recent research. *Tourism Management*, 29(2):203–220, 2008.

[19] Smita Sood and Keerti Jain. Comparative Analysis of Techniques for Forecasting Tourists' Arrival. *Journal of Tourism & Hospitality*, 06(03):2–6, 2017.

[20] Publisher Taylor, David A Dickey, Wayne A Fuller, David A Dickey, and Wayne A Fuller. Journal of the American Statistical Association Distribution of the Estimators for Autoregressive Time Series with a Unit Root Distribution of the Estimators for Autoregressive Time Series With a Unit Root. (July 2015):37–41, 2012.