 ISSN: 2394-7659
IMPACT FACTOR- 3.775

IJERMS

International Journal of Engineering Researches and Management Studies

TCS AND INFOSYS STOCK PRICES TIME-SERIES ANALYSIS AND PREDICTION USING THE ARIMA MODEL

Debaditya Raychaudhuri
Assistant Professor (WBES), Department of Computer Science, Bethune College, Govt. of West Bengal
Email: rana22021989@gmail.com

ABSTRACT

The IT and ITES industry has been the bedrock of India's economic and social growth for the last two decades. Today the Indian IT industry is not only one of the world's largest but it also represents the aspirations of India's burgeoning and vibrant middle class. It has been the pioneer in employment generation in the country. This is an impact of major Fortune 500 and Global 2000 multinational corporations outsourcing IT services to India. The IT industry has been a major contributor to India's GDP growth (both domestic and international export revenue). The IT sector has been a lucrative domain of stock market securities investment for quite a few years. We know that the Stock Exchange is an integral and important part of a Nation's economy. Two of India's largest IT companies are TCS (Tata Consultancy Services) and Infosys in terms of market capitalization, traded volume and employee strength. This paper attempts to analyze the weekly closing stock prices of TCS and Infosys for a span of more than 18 years, ranging from the 1st week of January, 2003 till the 3rd week of March, 2021. This paper also attempts to forecast the BSE weekly closing stock prices of TCS and Infosys for the next 5 years (260 weeks) at a weekly frequency. The paper also attempts to do an accuracy testing of the predictive model generated. This paper would be beneficial for both academic research purposes and for the individual investors who can use stock market predictions to make timely, informed and wise decisions to maximize their capital investments. The methodology employed in this paper for analyzing and forecasting stock prices of TCS and Infosys is Time Series Analysis using the ARIMA Model.

KEYWORDS: Arima model, Stock market prediction, Time Series analysis, IT stocks forecasting, Machine Learning.

1. INTRODUCTION

The IT and ITES industry has been the flamboyant backbone of the Indian economy for the last two decades. The IT industry basically provides general purpose technological, business and logistics solutions for scalability, mobility, production maximization, cost minimization, increase in throughput and efficiency quotients. The automation drive across sectors ranging from Bank & Financial Services to the Tourism industry to the Food delivery industry is built on the solutions and designs of the IT industry. This industry has enabled India's ever burgeoning middle class in realizing their "American Dream" sitting in India itself. The IT industry has been one of the largest employment generation sectors in India for the last 2 decades. Today a mammoth 4.36 million workforce is employed in the Indian IT industry. As on 31st December, 2019 the cumulative workforce strength of the 4 Indian IT majors (TCS, Infosys, Wipro and HCL Tech) is 1.02 million. Employment generation is the biggest social benefit this country has derived from the IT industry.

India has been the biggest outsourcing destination for major Fortune 500 and Global 2000 corporations for IT services. Out of the US$200-250 billion global services sourcing business estimated in FY 2019-20, India accounts for 55% of the market share. The IT and ITES industry's revenue generation was estimated at approx US$191 billion in FY 2019-20 and was growing at 7.7% year on year. The export revenues earned by the IT industry in FY 2019-20 was US$147 billion and the domestic revenues being US$44 billion. Thus it is evident that at a time when stagnation and an impending recession is hitting various sectors like Automobiles , Tourism etc. the IT industry continues to pull the nation ahead in terms of employment generation, foreign exchange earnings through exports and by maintaining a positive vibe in the Indian economy.

The stock market is an integral part of a country's economy. It provides a transaction facilitation centre for investors and traders to trade (buy/sell) securities of various companies enlisted in the Stock Exchange. The BSE (Bombay Stock Exchange) and the NSE (National Stock Exchange) are the two premier and prestigious stock exchanges of India.

Two of the largest Indian IT companies are TCS (Tata Consultancy Services) and Infosys. TCS was established in 1968. Today it is the largest IT firm in the world with a market capitalization of US$169.2 billion. It is the 1st Indian IT firm to reach the US$100 billion market capitalization. It is enlisted in BSE and NSE. TCS has a workforce of over 4.17 lacs.

Infosys was established in 1981. It has grown from being a US$250 firm to a US$72.2 billion firm today. It is enlisted in BSE and NSE. Infosys has a workforce of over 2.42 lacs.

Forecasting and predicting the troughs and crests of the stock market are difficult and challenging but highly essential. An assumption is made based on which stock prices forecasting have been done in this research paper. The data that is available in public domain (Yahoo Finance portal) has some relationship amongst them which has enabled to forecast the future trend of stock prices.

This paper will be beneficial for both academic research purposes and for individual investors seeking mechanisms for maximizing their profit-making from investments in renowned IT stocks. The foremost objective of this research work is development of a stock prices predictive model.

The data used in this paper is not a set of arbitrary, discontinuous data values. But it is a Timeseries data, which holds observations of variations of the same variable (weekly closing stock prices of TCS and Infosys) against a continuous and moving timeline collected at regular intervals (weekly in this case).

For building the stock prices predictive model, in this paper we have utilized the ARIMA (Auto-Regressive Integrated Moving Average) model for Timeseries analysis. It is a highly effective and robust statistical model for market fluctuations prediction. Here a non-stationary Timeseries is converted into a stationary one before it is used for predictive purposes. ARIMA methodology uses the Auto-Regression, Moving Average and Differencing statistical techniques.

The programming language used for the experiments conducted is R language. The IDE used is R GUI 64 bit.
In a nutshell, the research paper collects the weekly closing stock prices of TCS and Infosys from the 1st week of January, 2003 till the 3rd week of March, 2021 (948 weeks). Exploratory Data Analysis (EDA) is done on the aforementioned data. Based on this data a predictive model is generated using ARIMA. The model is used for forecasting the weekly closing stock prices of TCS and Infosys for the next 5 years (260 weeks). It is also used for accuracy checking in terms of Mean Absolute Percentage Error (MAPE) and Mean Percentage Error (MPE) by segregating the complete dataset into Training and Testing datasets.

Section 1 deals with a short introduction about the research work. Section 2 declares the objectivity and aim of the study. Section 3 enlists a set of related work in this area of research. Section 4 speaks about the theory, statistics and literature associated with the ARIMA model. Section 5 describes the data source and the data used in the experiment and Section 6gives a diagrammatic representation of the methodology employed in this paper. Section 7 gives a detailed account of the experiment results achieved in this experiment and Section 8 concludes the research work.

2. OBJECTIVE

As already discussed, the foremost objective of this research paper is to analyze the BSE weekly closing stock prices data of TCS and Infosys from the 1st week of January, 2003 (01.01.2003) till the 3rd week of March, 2021 (19.03.2021). The weekly data collected in this case is the closing stock price value recorded at the end of trading sessions on each Wednesday (Wednesday being chosen because of it being in the middle of the week) of the aforementioned period. Then a predictive forecasting model is built by utilizing the ARIMA modeling technique. This predictive model is used for forecasting weekly closing stock prices of TCS and Infosys from the 4th week of March, 2021 till the 3rd week of March, 2026 (five years or 260 weeks). Then the complete dataset is divided into Training and testing datasets for predictive accuracy calculation purposes.

3. RELATED WORK

There are several notable and praiseworthy research publications in this field. The authors Mohankumari C, Vishukumar M and Nagaraja Rao Chillale had developed a predictive model for forecasting stock prices using the ARIMA model [1]. They has used ARIMA models for analyzing and forecasting NSE stock prices of TCS, Infosys, HCL , Tech Mahindra and Wipro. For each of the different aforementioned firms, the authors designed different Arima models due to different optimal values of p,q and d. The authors B. Uma Devi, D.Sundar and Dr. P. Alli had used AR and MA models along with AIC and BIC scores for analyzing the stock indices of NIFTY-50 Midcap, Reliance, OFSS, JSW-Steel and ABB [2]. They also did a accuracy check on the results obtained using the AR and MA models. The author Nazmul Islam used the ARIMA model for forecasting the inflation rate of Bangladesh [3]. He used the World Bank inflation data of Bangladesh ranging from 1971 till 2015 (1973 to 1978 were ignored due to outlier problem) to forecast the inflation rate of the country from 2016 to 2020. He chose the ARIMA parameters based on the minimum AIC value. The authors Aloysius Edward and Jyothi Manoj had developed a predictive model for forecasting the stock prices of various premier automobile sector stocks like Mahindra & Mahindra, Bajaj Auto, Hero Motors and Tata Motors [4]. He converted non-stationary series into stationary one before using ACF and PACF graphs for designing the ARIMA model for prediction. The authors Mohamed Ashik A and Senthamarai Kannan K used the ARIMA model for forecasting the closing stock indices of NIFTY-50 based on the Nifty historical indices data ranging from January to December 2015 (245 observations) [5]. The authors Mahantesh C. Angadi and Amogh P. Kulkarni did exploratory data analysis on intra-day Infosys stock prices data ranging from 2007 to 2015 and developed an ARIMA predictive model using various parameter values for p.q and d [6]. The authors Kamalakannan J, Indrani Sengupta and Snehaa Chaudhury had retrieved Apple

Inc stock prices of the period October 2015 till October 2017 from Yahoo Finance and performed ARIMA modeling technique on the data to predict the future stock prices for a period of 2 years [7]. The authors Ayodele A. Adebiyi , Aderemi O. Adewumi , Charles K. Ayo used historical stock prices of Nokia and Zenith Bank from the New York stock Exchange and Nigeria Stock Exchange to design predictive models for forecasting future stock prices. They used the ARIMA modeling technique with Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) [8]. The authors Nitin Merh, Vinod P. Saxena and Kamal Raj Pardasani used the historical prices of BSE 30 (SENSEX), BSE IT, BSE Oil & Gas, BSE 100 and S&P CNX Nifty for future price prediction [9]. They developed 2 types of hybrid models by combining the ARIMA and ANN models. Using the hybrid models they generated predictions.

4. LITERATURE RELATED TO THE ARIMA MODEL

The ARIMA (Auto-Regressive Integrated Moving Average) Model is a mathematical, statistical model used for analysis of a Time-series and prediction of future values of the Time-series based on the past values of it. A timeseries is a sequence of numerical data observed in regular intervals of time over a span of time. The regular interval is called the frequency of the timeseries. In this research paper , we shall analyze an uni-variate timeseries.In an uni-variate timeseries , the present value of the dependant variable is a function of its past values.

Here, $y_t = y_{t-1} + Error/Randomness$ (1)

In Equation (1), y_t is the dependant variable at time instant 't' which depends upon the value of y at time instant 't-1' given by y_{t-1} .Some common and classical examples of timeseries are – monthly average airplane passengers, weekly closing stock prices, yearly inflation data etc. The principal components of a Time Series are – *Trend, Seasonality, Cyclicity* and *Randomness/Error.* A time series can be decomposed into its constituent components – trend, seasonality and randomness. The decomposition can be classified into two categories: Additive decomposition and Multiplicative decomposition.

Additive decomposition:
$Y_t = Trend_t + Seasonality_t + Randomness_t$ (2)
Multiplicative decomposition:
$Y_t = Trend_t * Seasonality_t * Randomness_t$ (3)

In an *Auto Regression (AR) model*, we predict the uni-variate variable using a linear function of past values of the variable. The term *autoregression* denotes that it is a regression of the variable against its own values. Thus, an autoregressive model of order *'p'* can be written as - $\mathbf{y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + \cdots + a_p * y_{t-p} + e_t,}$ (4)

where $\mathbf{e_t}$ is white noise. This is similar to a multiple regression but with *lagged values* of y_t as predictors. This shall be referred to as an **AR(p) model**, an autoregressive model of order p.
AR(1) : $y_t = a_0 + a_1 * y_{t-1} + e_t$ (5)
AR(2) : $y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + e_t$ (6)
So, in order to represent y(t) in a AR model of order *'p'* in terms of a linear function we can write :
$\mathbf{Y_t = F(y_{t-1} , y_{t-2}, \ldots, y_{t-p} , e_t)}$ (7)

In a *Moving Average (MA) model* ,instead of using past values of the predicted variable in a regression, a moving average model uses past predicted errors (white noise) in a regression-type model. Thus, an moving average model of order *'q'* can be written as –
$\mathbf{y_t = b_0 + e_t + b_1 * e_{t-1} + b_2 * e_{t-2} + \cdots + b_q * e_{t-q}}$ (8)
where $\mathbf{e_t}$ is white noise. This shall be referred to as an **MA(q) model**, a moving average model of order q.
MA(1) : $y_t = b_0 + e_t + b_1 * e_{t-1}$ (9)
MA(2) : $y_t = b_0 + e_t + b_1 * e_{t-1} + b_2 * e_{t-2}$ (10)
So, in order to represent y_t in a MA model of order *'q'* in terms of a linear function we can write :
$\mathbf{Y_t = F(e_{t-1} , e_{t-2} , \ldots, e_{t-q} , e_t)}$ (11)
Then after deriving the AR and MA models we need to combine the models. We get the *Auto Regression Moving Average (ARMA) model*. It is a combination of AR(p) and MA(q) , denoted by ARMA(p,q) . Here y(t) is a linear function of the past values of y and also the past white noise(error) values. Thus, an ARMA model of order *' p,q'* can be written as –
$\mathbf{y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + .. + a_p * y_{t-p} + e_t + b_0 + b_1 * e_{t-1} + b_2 * e_{t-2} + ... + b_q * y_{t-q} + e_t}$ (12)

ARMA (0, 1): $y_t = a_0 + b_0 + e_t + b_1 * e_{t-1}$ (13)
ARMA (1, 0): $y_t = a_0 + a_1 * y_{t-1} + b_0 + e_t$ (14)
ARMA (1, 1): $y_t = a_0 + a_1 * y_{t-1} + b_0 + b_1 * e_{t-1} + e_t$ (15)

A stationary time series is a series whose statistical attributes such as mean, variance, are all constant over time. Most

IJERMS

statistical predictive methods are based on the assumption that the time series can be rendered approximately stationary (i.e., "stationarized") through the use of mathematical/ statistical transformations like differencing. A time series can be used for predictive purposes using the ARIMA if and only if it is a stationary series. Stationarity of a Time Series can be tested or verified using the Augmented Dickey-Fuller (ADF) test .To check stationarity perform ADF test for the null hypothesis of a unit root of a single variable (univariate) time series. Here, null hypothesis is "not stationary" and alternate hypothesis is "stationary".

One way to convert a non-stationary time series into a stationary one is to compute the differences between 2 consecutive observations or values. This is known as Differencing. It can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality. Differencing converts a non-stationary series into a stationary one. We can say that, differencing de-trends a time series, i.e. it removes the trend component. The differenced time-series is the alteration between consecutive observations in the original time-series, and can be represented as $y'_t = y_t - y_{t-1}$. The differenced time-series will have only T−1 values, since it is not possible to calculate a difference y'_1 for the first observation or value of the time-series. The number of times differencing is computed on the series is called the "order of differencing". Denoted by'd'.

1st order differencing: $y'_t = y_t - y_{t-1}$ (16)

The differenced series will have only T−1 values, because it is not possible to calculate a difference y'_1 for the first observation.

2nd order differencing:

$$y''_t = y'_t - y'_{t-1} \quad (17)$$

$$y''_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \quad (18)$$

$$y''_t = y_t - 2*y_{t-1} + y_{t-2}. \quad (19)$$

In this case, y''_t will have T−2 values.

Earlier when we combined the AR and the MA models then we had derived the ARMA model. Now we combine it with the Differencing that was essential for achieving the stationarity . It is called as the Integrated model. When we combine the 3 models we get the ARIMA model. It is a combination of AR(p) , I(d) and MA(q) , denoted by ARIMA(p,d,q) . Here yt is a function of the past values of y and also the past white noise(error) values. Thus, an ARIMA model of order 'p , d , q' can be written as –

$$y'^{d}_t = a_0 + a_1*y_{t-1} + a_2*y_{t-2} + .. + a_p*y_{t-p} + e_t + b_0 + b_1*e_{t-1} + b_2*e_{t-2} + .. + b_q*y_{t-q} + e_t \quad (20)$$

For a time series, the Partial Auto-Correlation Function (PACF) between xt and xt-h can be defined as the conditional correlation between xt and xt-h , conditional on the values of xt-h+1, ... , xt-1, the set of observations (values) that come between the time instants 't' and 't−h'. At lag 'h', this is the correlation between the time-series observations (values) that are 'h' intervals apart, accounting for the values of the intervals between. The order of PACF is 'p' which is the required parameter for the AR model.

For a time series, AutoCorrelation Function (ACF) between xt and xt-h be defined as the correlation between xt and xt-h . At lag 'h', this is the correlation between the values of xt and xt-h without taking into consideration the interval values (as was the case in PACF). The order of ACF is 'q' which is the required parameter for the MA model. Using the parameters p, d and q the ARIMA model is generated which will be used for predictive and forecasting purposes and accuracy checking.

5. DATA SOURCE USED FOR THE RESEARCH

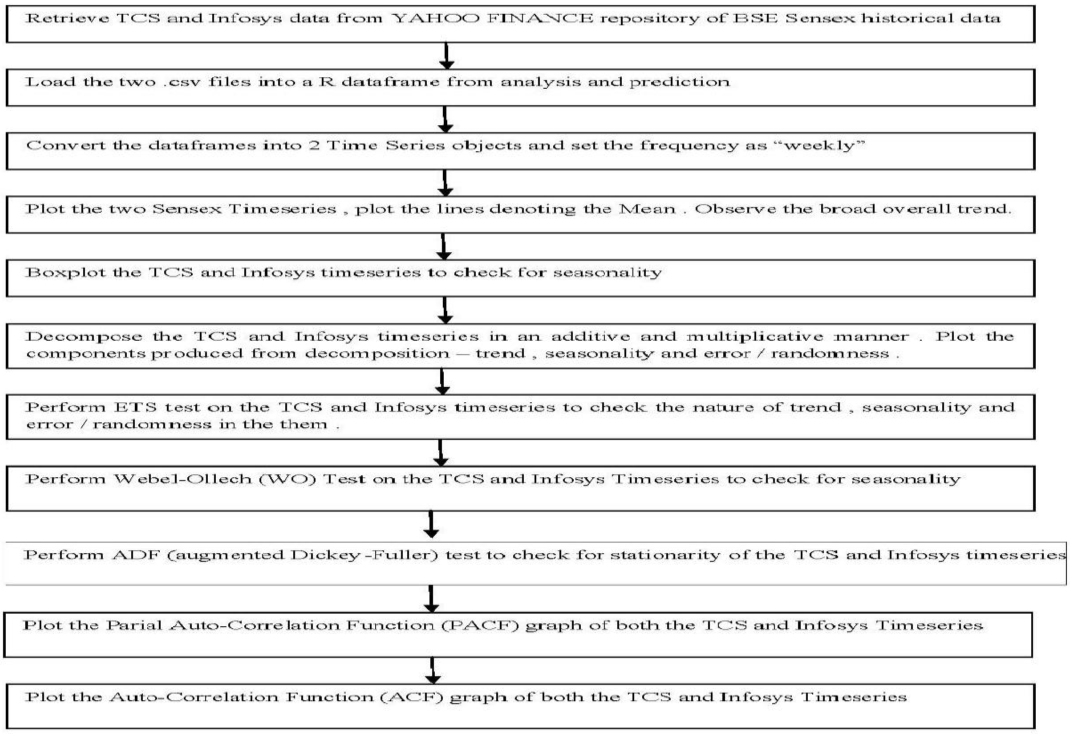
The datasets used for the experiments recorded in this research paper have been retrieved from YAHOO FINANCE portal holding TCS and Infosys historical data of stock prices. Two separate CSV files – one for TCS stock prices data and another for Infosys stock prices data have been retrieved. Each of these CSV files has 948 rows (denoting 948 weeks from 1st week of January, 2003 till the 3rd week of March, 2021) and 7 columns (date, opening value, high, low, closing value, adjusted closing value, traded volume). The paper uses the "Closing" column for analysis and predictive purposes. The CSV files are converted into data-frames in R.

6. METHODOLOGY OF THE EXPERIMENT

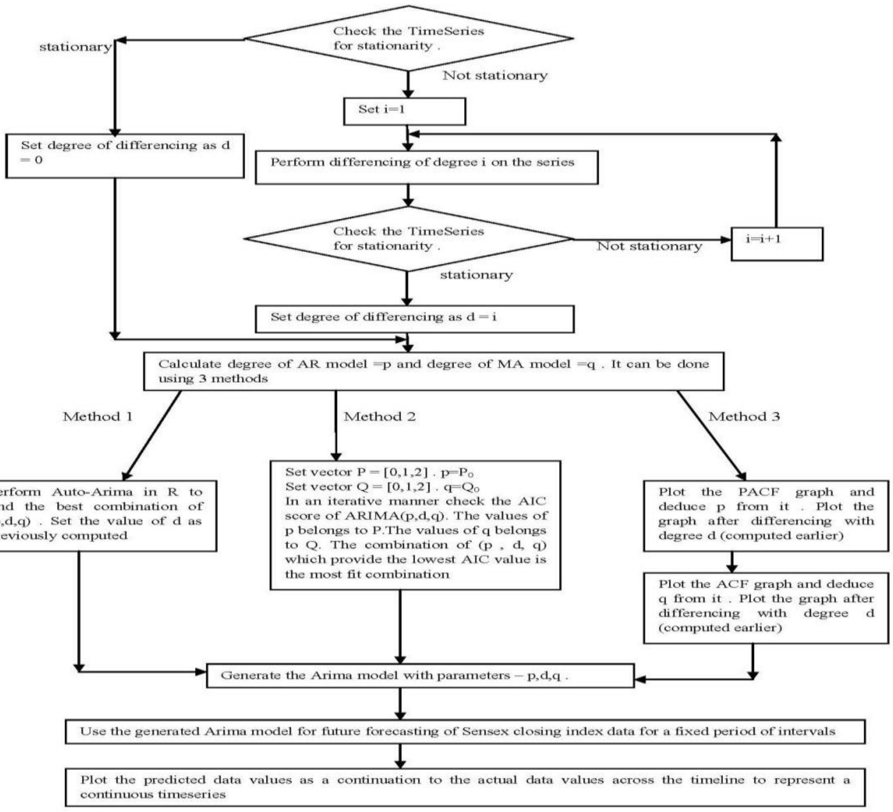
The experiment can be broadly divided into 3 phases – Data Exploratory Analysis, Forecasting Using the Predictive Model and Accuracy Testing Using the Trained Model.

6.1 Data Exploratory Analysis

ISSN: 2394-7659
IMPACT FACTOR- 3.775

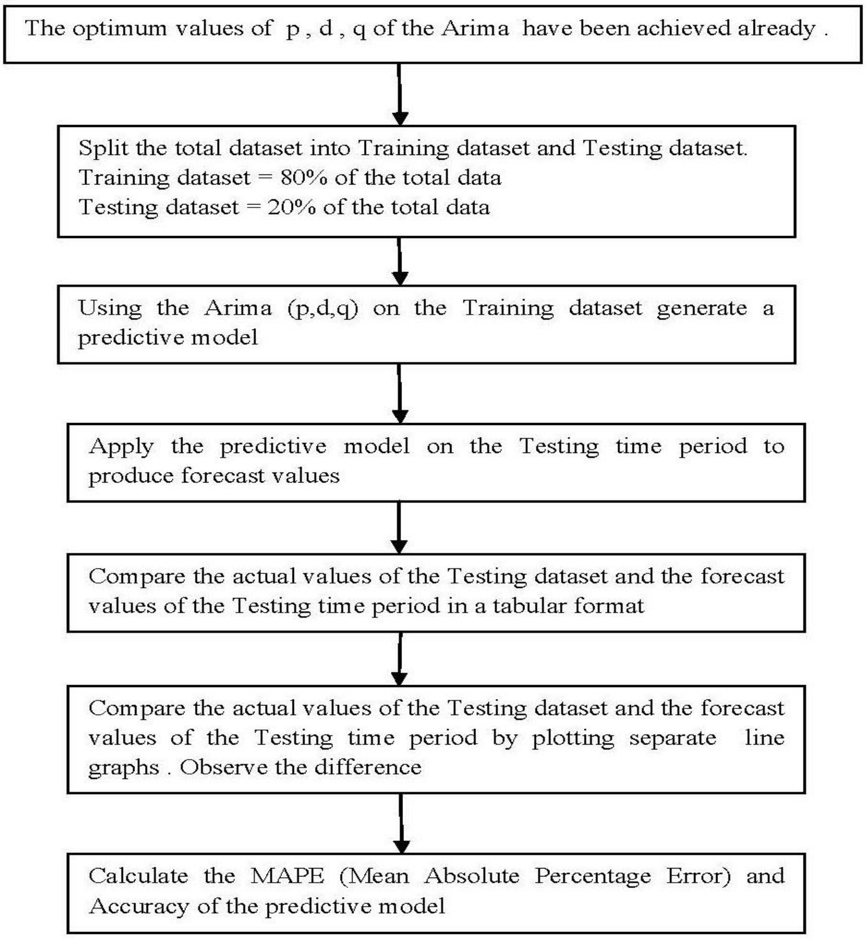


6.2 Forecasting Using The Predictive Model (Perform the following steps for both TCS and Infosys Timeseries)



6.3 Accuracy Testing Using The Trained Model (Perform the following steps for both TCS and Infosys Timeseries)

ISSN: 2394-7659
IMPACT FACTOR- 3.775



7. RESULTS AND DISCUSSION

The experiments performed in this paper have been done using the R language and RGUI 64 bit IDE has been used. The data is retrieved from the YAHOO Financial data repository which holds the historical prices of all the BSE and NSE registered stocks. In this case, 2 sets of data have been retrieved. One is the TCS stock prices data from 1st week of January, 2003 till the 3rd week of March, 2021 (weekly data observed at close of trade on every Wednesday). Second data is the INFOSYS stock prices data from 1st week of January, 2003 till the 3rd week of March, 2021 (weekly data observed at close of trade on every Wednesday). Both these data sets have 948 observations (denoting 948 weeks of observations). The snapshot of a portion of the total TCS data retrieved is given below in Fig. 1. The snapshot of a portion of the total INFOSYS data retrieved is given below in Fig. 2:

```
> MyData <- read.csv(file="F:/PAPER IT STOCKS/PROGRAM/TCS.NS.csv",header=TRUE, $
> MyData
        Date    Open    High     Low   Close Adj.Close   Volume
1   1/1/2003 59.9875 62.9250 56.0625 56.6000  42.38336 22322760
2   1/8/2003 57.3500 60.1500 54.2500 56.8875  42.59865 11642528
3  1/15/2003 57.3500 60.2000 54.0000 54.2375  40.61429  8962672
4  1/22/2003 54.7000 55.7500 46.2500 50.3625  37.71258  6480128
5  1/29/2003 50.5250 53.4750 48.1750 50.5500  37.85298  6534896
6   2/5/2003 49.7500 51.9125 47.7500 48.0250  35.96221  3966792
7  2/12/2003 48.5250 49.0000 44.5625 47.5625  35.61588  2864360
8  2/19/2003 48.2500 48.4625 45.3875 45.8000  34.29608  2084376
9  2/26/2003 46.2750 48.9875 44.6500 46.3375  34.69857  2154136
10  3/5/2003 45.8250 47.0000 39.5750 42.0125  31.45992  2098864
11 3/12/2003 43.1750 43.1750 40.2750 41.8000  31.30078  1245840
12 3/19/2003 42.8750 44.2250 40.1250 40.9875  30.69236  1774160
13 3/26/2003 42.2500 42.2500 32.4750 38.3875  28.74543   950032
14  4/2/2003 38.3875 38.3875 38.3875 38.3875  28.74543        0
15  4/9/2003 38.3875 38.3875 38.3875 38.3875  28.74543        0
```

Fig. 1. A snapshot of a portion of the whole TCS data

```
> MyData <- read.csv(file="F:/PAPER IT STOCKS/PROGRAM/INFY.BO.csv",header=TRUE,$
> MyData
         Date     Open     High      Low    Close Adj.Close    Volume
1    1/1/2003 74.40625 76.14062 73.59375 74.33203  53.34440 120092032
2    1/8/2003 74.60938 75.28125 66.56250 69.51172  49.88511 173196800
3   1/15/2003 70.31250 72.91563 69.73438 69.92344  50.18057 122840128
4   1/22/2003 69.92188 71.11172 66.56250 67.33203  48.32084  95047488
5   1/29/2003 67.81250 70.22969 65.29531 68.96641  49.49377  90743168
6    2/5/2003 68.43750 71.07812 66.06250 66.33672  47.60656  91362560
7   2/12/2003 66.39062 67.18750 64.37500 66.26094  47.55217 101867328
8   2/19/2003 66.85938 67.70547 64.30469 64.47266  46.26881  88985792
9   2/26/2003 64.53125 67.87500 63.46875 64.94063  46.60466  96474816
10   3/5/2003 64.53125 65.60938 62.26562 64.52266  46.30471  92201472
11  3/12/2003 64.23828 64.79688 61.82812 63.09531  45.28036  62523584
12  3/19/2003 63.28125 69.82812 63.28125 66.65859  47.83754 117076928
13  3/26/2003 66.64062 67.59375 62.34375 63.75547  45.75413  78904064
14   4/2/2003 63.91406 68.73281 63.12500 66.04297  47.39575  91860416
15   4/9/2003 65.62500 65.78125 37.81250 44.72656  32.09803 321635456
16  4/16/2003 45.15469 47.57265 43.92188 45.66250  32.76970 135620608
17  4/23/2003 46.09375 46.84375 44.37578 44.66953  32.05710  87481600
18  4/30/2003 44.64062 47.95312 43.03125 46.67812  33.49856  96362816
19   5/7/2003 47.26562 47.87500 43.09375 44.90234  32.22419  92654592
20  5/14/2003 44.76562 47.60938 42.65703 44.71719  32.09130 116178240
21  5/21/2003 44.84375 45.70312 40.18750 40.70625  29.21285 112315328
```

Fig. 2. A snapshot of a portion of the whole INFOSYS data

For experimental purposes only the "Close" attribute is used for analysis and prediction. So, the datasets are preprocessed to retrieve only the required attribute (Close) along with the Date. Then the datasets are translated into a two Timeseries called "Tcs" and "Infosys"(frequency being weekly) . Snapshots of the processed Timeseries datasets (TCS and INFOSYS) are given below in Fig. 3 and Fig. 4:

```
> tcs <- ts(MyData$Close, start=c(2003,1), freq=52 )
> tcs
Time Series:
Start = c(2003, 1)
End = c(2021, 12)
Frequency = 52
  [1]  56.6000  56.8875  54.2375  50.3625  50.5500  48.0250  47.5625
  [8]  45.8000  46.3375  42.0125  41.8000  40.9875  38.3875  38.3875
 [15]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [22]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [29]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [36]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [43]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [50]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [57]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875
 [64]  38.3875  38.3875  38.3875  38.3875  38.3875  38.3875 105.3125
 [71]  80.0375  86.0000  80.1125  77.4750  86.8125  85.9750  89.2375
 [78]  89.2125  88.3375  97.1875  96.9375  99.9000 101.2000 102.4125
 [85] 115.1750 123.5125 124.4500 126.9575 130.7250 126.9450 135.0875
 [92] 135.5575 142.9000 142.1500 144.9625 148.5500 148.8625 154.4450
 [99] 159.3325 152.2825 154.1450 159.0875 163.6700 167.6575 164.2200
[106] 157.9825 154.4325 164.4075 164.5000 172.6700 170.4325 169.6825
[113] 176.7450 180.3325 180.4450 171.3625 178.1825 172.7500 151.0875
[120] 142.0875 138.5750 144.0625 143.8200 154.3575 166.7825 159.7075
```

Fig. 3. A snapshot of the total Timeseries "TCS" dataset from 1st week of January, 2003 till the 3rd week of March, 2021

```
> infy <- ts(MyData$Close, start=c(2003,1), freq=52 )
> infy
Time Series:
Start = c(2003, 1)
End = c(2021, 12)
Frequency = 52
  [1]  74.33203  69.51172  69.92344  67.33203  68.96641  66.33672
  [7]  66.26094  64.47266  64.94063  64.52266  63.09531  66.65859
 [13]  63.75547  66.04297  44.72656  45.66250  44.66953  46.67812
 [19]  44.90234  44.71719  40.70625  42.51718  44.89062  48.38281
 [25]  49.12500  51.63984  50.64453  53.19219  52.30234  56.10859
 [31]  58.31484  53.19531  54.50469  58.38203  60.63281  70.11328
 [37]  65.57422  67.41719  70.69063  69.62578  73.65625  71.13672
 [43]  69.92188  78.23203  77.33125  72.10000  73.96016  80.41172
 [49]  77.74766  81.76328  85.31406  88.35390  90.52812  84.24453
 [55]  87.36406  81.46172  86.04375  84.39297  78.75547  83.15938
 [61]  77.66093  77.33125  80.58594  78.71719  83.00234  85.78438
 [67]  84.06563  80.23985  81.50625  78.98203  76.35547  83.73515
 [73]  82.61094  80.97734  79.44297  83.99922  87.30312  89.14375
 [79]  87.65312  90.47813  92.42812  98.55000  95.23438  92.59062
 [85]  97.25000  98.43125 101.02812 106.09688 104.76250 100.71250
 [91] 107.34062 107.01562 110.41875 116.41563 119.37813 127.20312
 [97] 128.05000 126.92188 134.27188 127.43437 128.20000 132.24374
[103] 128.86563 130.52188 123.18437 122.70000 119.49375 128.97501
[109] 127.56875 136.50313 135.11563 136.35312 141.51250 137.80937
```

Figure 4. A snapshot of the total Timeseries "INFOSYS" dataset from 1st week of January, 2003 till the 3rd week of March, 2021

Data exploratory analysis is performed by plotting the TCS and INFOSYS timeseries graph along with its mean line to observe the trend. There is a general upward increasing trend that can be visualized in both the timeseries datasets. The graphical representation of the TCS and INFOSYS timeseries and its mean line are given below in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

ISSN: 2394-7659
IMPACT FACTOR- 3.775

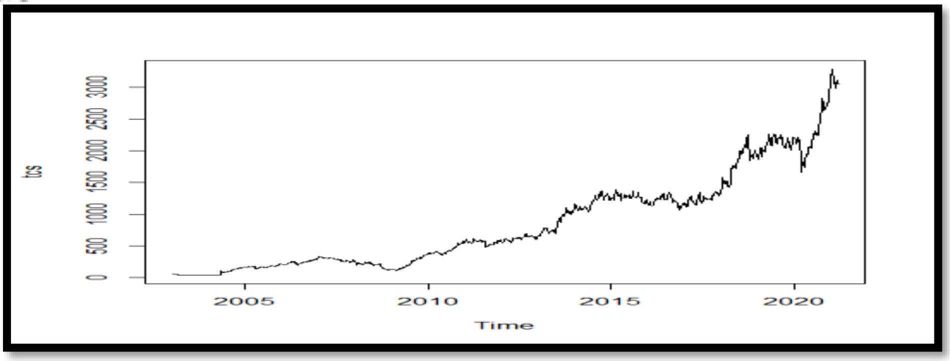


Fig. 5. Graph representing the TCS Timeseries data

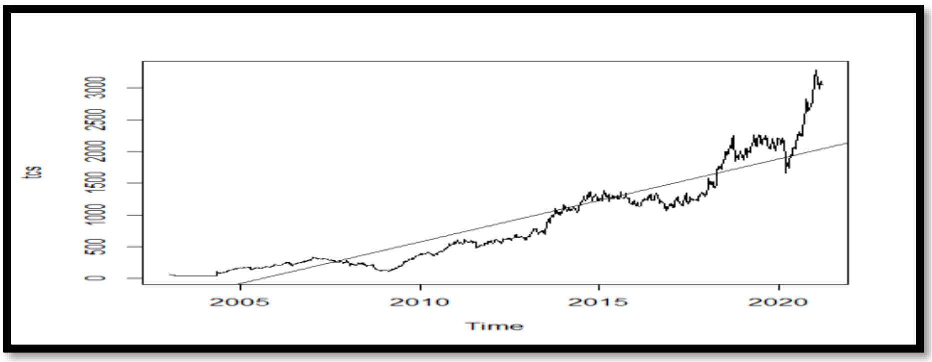


Fig. 6. Graph representing the TCS Timeseries data along with its Mean line

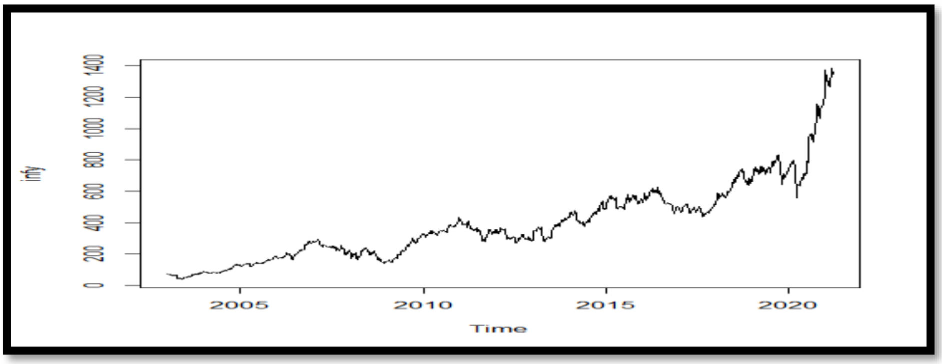


Fig. 7. Graph representing the INFOSYS Timeseries data

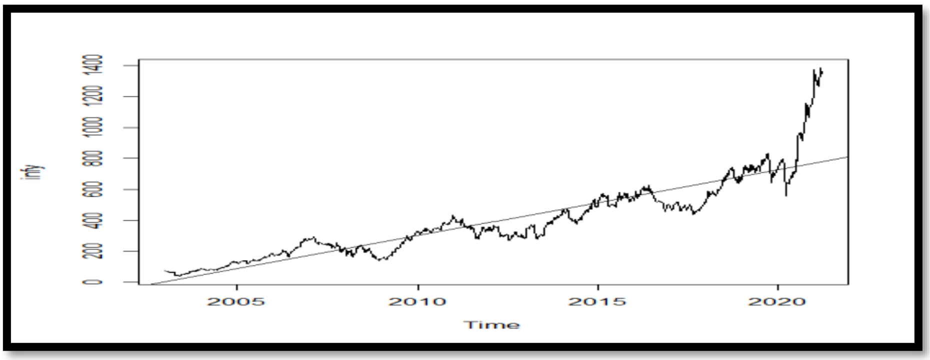


Fig. 8. Graph representing the INFOSYS Timeseries data along with its Mean line

The Boxplot curves for the TCS and Infosys datasets are also plotted. The boxplots are given in Fig. 9 and Fig. 10.

ISSN: 2394-7659
IMPACT FACTOR- 3.775

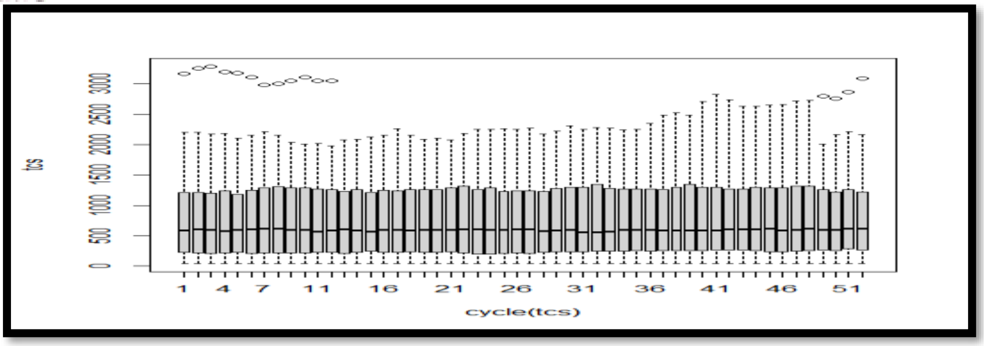


Fig. 9. Boxplot representing the TCS Timeseries data

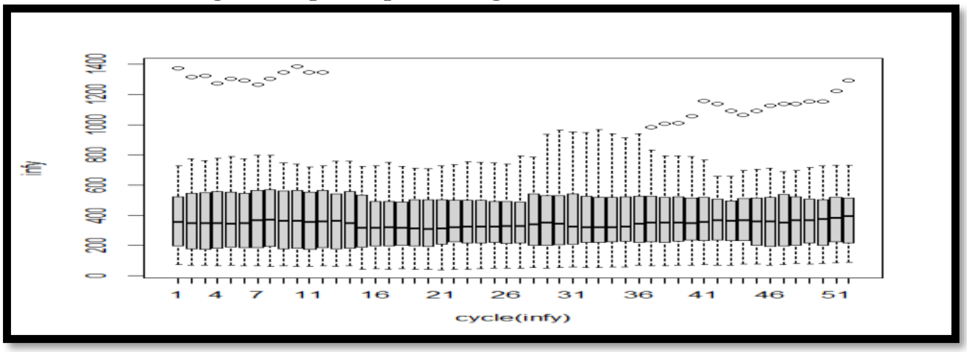


Fig. 10. Boxplot representing the INFOSYS Timeseries data

The TCS and Infosys timeseries(s) are decomposed into their principal components – Trend, Seasonality and Randomness / Error. The graphical representation the decomposed components of the aforementioned datasets are given below in Fig. 11 and Fig. 12.

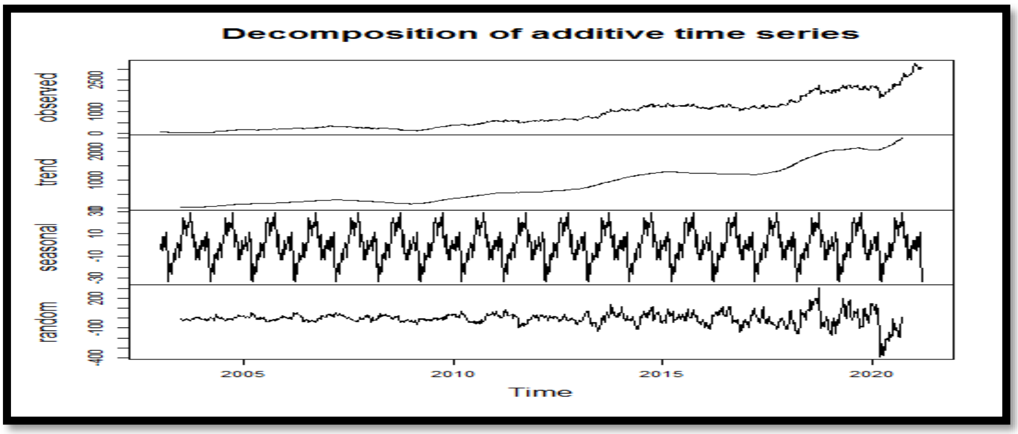


Fig. 11. Graph denoting the decomposed components of the TCS dataset

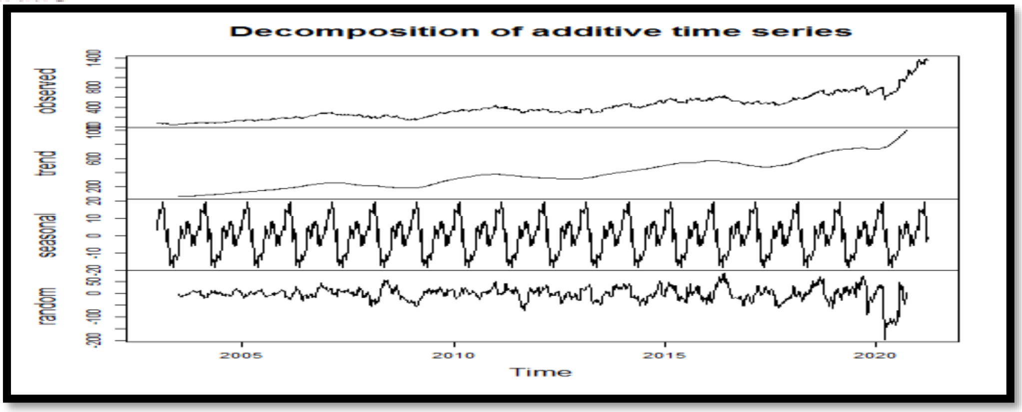


Fig. 12. Graph denoting the decomposed components of the INFOSYS dataset

The ETS (Error , Trend , Seasonality) test is performed on the TCS and INFOSYS timeseries . The results obtained are given below in Table 1 -

Data	Error	Trend	Seasonality
TCS Data	Multiplicative	Additive	None
INFOSYS Data	Multiplicative	Additive	None

Table 1. ETS Test Results

The ADF (Augmented Dickey-Fuller) test is performed on the TCS and INFOSYS timeseries to check for stationarity . The confidence level is taken to be 95% (0.95). The results obtained are given below in Table 2 –

Data	T-Statistic	P-Value	Conclusion
TCS Data	0.093481	0.99	Not Stationary
INFOSYS Data	0.93936	0.99	Not Stationary

Table 2. ADF Test Results

For checking the Seasonality of the timeseries datasets, we have used the Webel and Ollech (WO) Test. The WO test produces 3 separate results – QS , QS-R and KW-R test results. The WO Test results are given below in Table 3-

Data	T-Statistic	P-Value			Conclusion
		QS-test	QS-R test	KW-R-test	
TCS Data	0	1	1	0.9275484	Not Seasonal
INFOSYS Data	0	0.5215291	0.7003536	0.1141231	Not Seasonal

Table 3. Webel and Ollech (WO) Test Results for checking Seasonality

From the ETS Test , ADF Test and WO Test results given in Table 1, Table 2 and Table 3 respectively, we can conclude that both the datasets are Not Seasonal in nature. But, the ETS and ADF Test results indicate that both the TCS and INFOSYS datasets are Not Stationary in nature. Thus in order to generate predictive models using these datasets we need to stationarize these datasets. Both the datsets have an upward trend. Usind d=1, one degree of differencing is done on both the datasets . The results are given below in Fig.13, Fig. 14, Fig. 15, Fig. 16, Fig. 17 and Fig. 18:

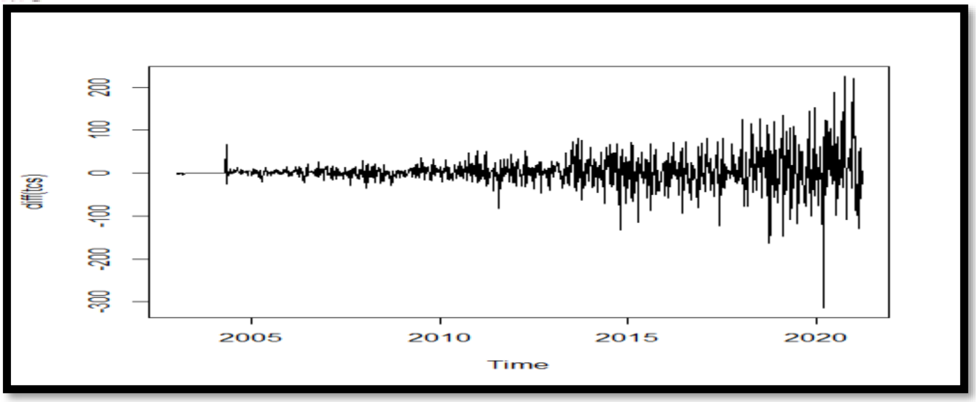


Fig. 13. Graph representing the TCS Timeseries data with Differencing d=1

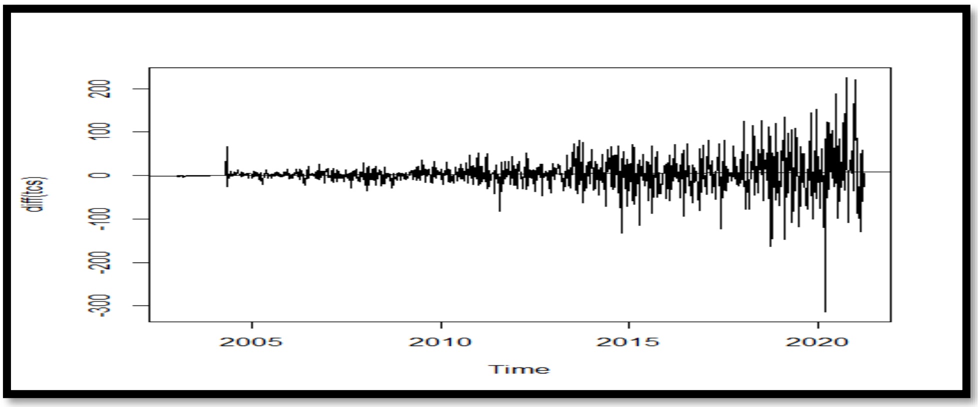


Fig. 14. Graph representing the TCS Timeseries data and its Mean Line with Differencing d=1

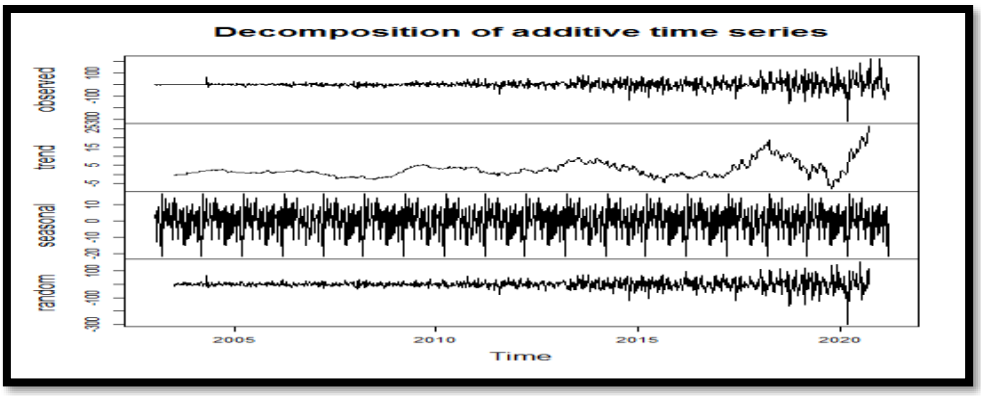


Fig. 15. Graph denoting the decomposed components of the TCS dataset with Differencing d=1

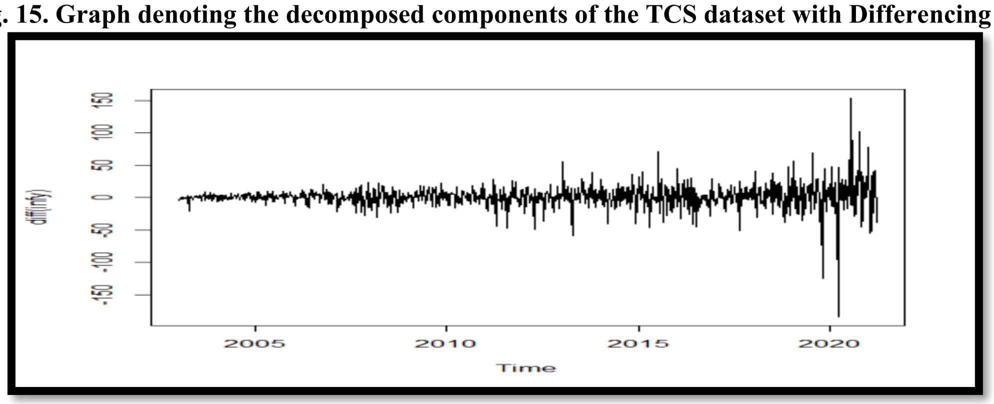


Figure 16. Graph representing the INFOSYS Timeseries data with Differencing d=1

ISSN: 2394-7659
IMPACT FACTOR- 3.775

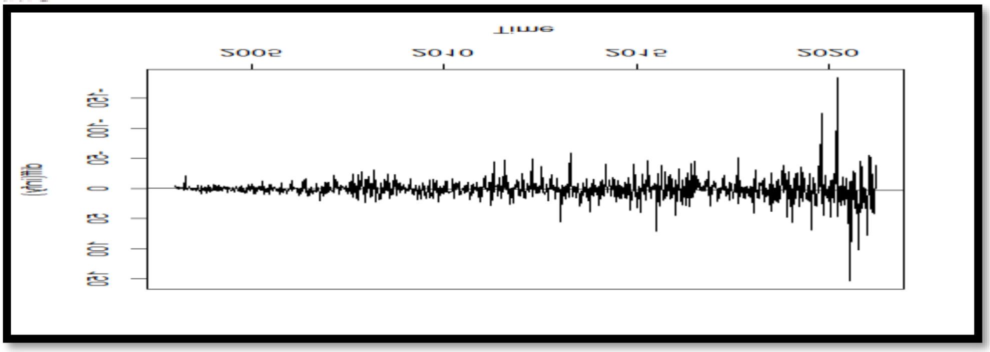


Figure 17. Graph representing the INFOSYS Timeseries data and its Mean Line with Differencing d=1

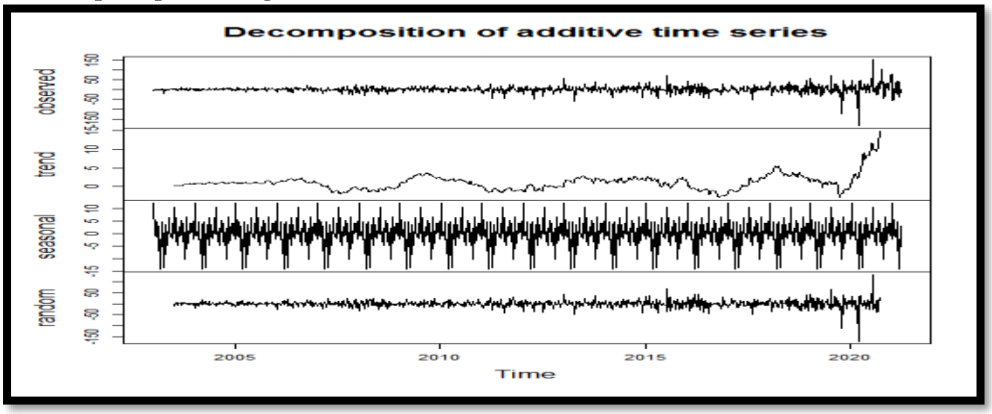


Fig. 18. Graph denoting the decomposed components of the INFOSYS dataset with Differencing d=1

It is evident from the Fig.13, Fig. 14, Fig. 15, Fig. 16, Fig. 17 and Fig. 18 graphs that the upward trend has dissappeared after the differencing step. Now , the ETS test is performed on the differenced timeseries. The results obtained are given below in Table 4 –

Data	**Error**	**Trend**	**Seasonality**
TCS Data after Differencing (d=1)	Additive	None	None
INFOSYS Data after Differencing (d=1)	Additive	None	None

Table 4. ETS Test Results after Differencing (d=1)

The ADF (Augmented Dickey-Fuller) test is performed on the differenced timeseries to check for stationarity . The confidence level is taken to be 95%(0.95) . The results obtained are given below in Table 5 –

Data	**T-Statistic**	**P-Value**	**Conclusion**
TCS Data after Differencing (d=1)	-10.091	0.01	Stationary
INFOSYS Data after Differencing (d=1)	-9.7035	0.01	Stationary

Table 5. ADF Test Results after Differencing (d=1)

As is evident from the results of the ETS and the ADF tests , the TCS and INFOSYS timeseries after 1 step differencing becomes stationary . So , the value of d of the ARIMA model has been set to 1. To compute the values of p and q , 3 methods were discussed in the earlier section. Now they will be implemented.

Method 1 : Using the Auto Arima functionality of R , the best combination of ARIMA (p,d,q) for TCS timeseries data comes out to be : ARIMA(0,1,2) with drift . Using the Auto Arima functionality of R , the best combination of ARIMA (p,d,q) for INFOSYS timeseries data comes out to be : ARIMA(3,2,1) with drift .

Method 2 : Using the iterative process of selecting the optimal values of p and q (set d=1) for the lowest AIC (Akaike Information Criterion) value for TCS timeseries data gives the following result in Table 6 –

q / p	q=0	q=1	q=2
p=0	9575.829	9571.743	9565.173
p=1	9572.857	9564.468	9564.903
p=2	9566.143	9565.819	**9564.241**

Table 6. Iterative Test Results for determining value of p and q using AIC of TCS Data

The AIC value given by p=2 , q=2 (ARIMA(2,1,2)) is the lowest . Thus using this method the function is ARIMA (2,1,2) for the TCS timeseries data.
Using the iterative process of selecting the optimal values of p and q (set d=1) for the lowest AIC (Akaike Information Criterion) value for INFOSYS timeseries data gives the following result in Table 7 –

q / p	q=0	q=1	q=2
p=0	**8142.813**	8144.722	8145.435
p=1	8144.729	8146.633	8146.454
p=2	8145.287	8146.285	8145.152

Table 7. Iterative Test Results for determining value of p and q using AIC of INFOSYS Data

The AIC value given by p=0 , q=0 (ARIMA(0,1,0)) is the lowest . Thus using this method the function is ARIMA (0,1,0) for the INFOSYS timeseries data.

Method 3 : Plot the PACF and the ACF graphs and deduce the vlues and p and q from it. The PACF graph and the ACF graph for the TCS timeseries data has been plotted and are given in Fig. 19 and Fig. 20.

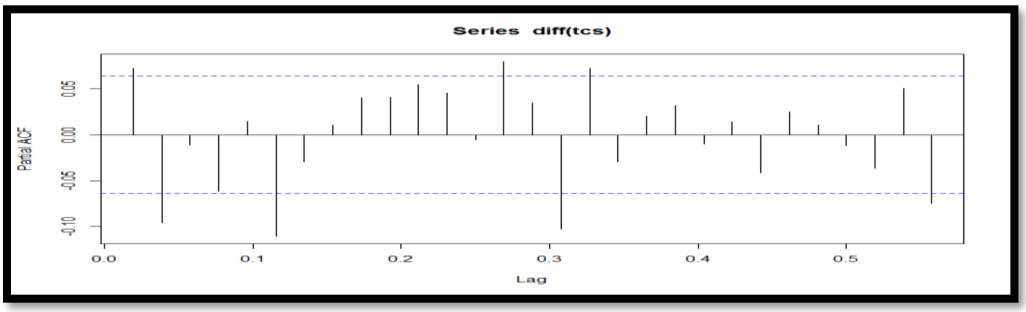


Figure 19. Figure representing the PACF graph of the TCS timeseries data

It is evident from the PACF graph in Fig. 19 , that after the spike at lag=2, the graph cuts off and thus decreases the correlation factor after lag=2 . So p=2 is deduced from the PACF graph.

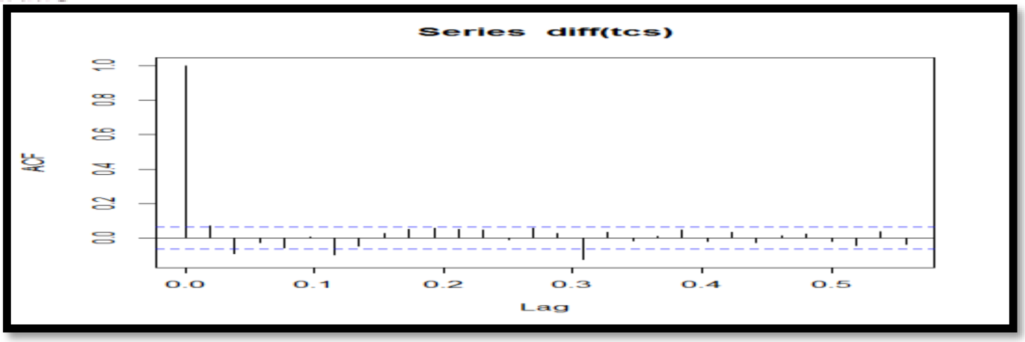


Figure 20. Figure representing the ACF graph of the TCS timeseries data

It is evident from the ACF graph in Fig. 20 , that after the spike at lag=2, the graph cuts off and thus decreases the correlation factor after lag=2 . So q=2 is deduced from the PACF graph.

Thus using this method the function is ARIMA (2,1,2) for the TCS timeseries data.
The PACF graph and the ACF graph for the INFOSYS timeseries data has been plotted and are given in Fig. 21 and Fig. 22.

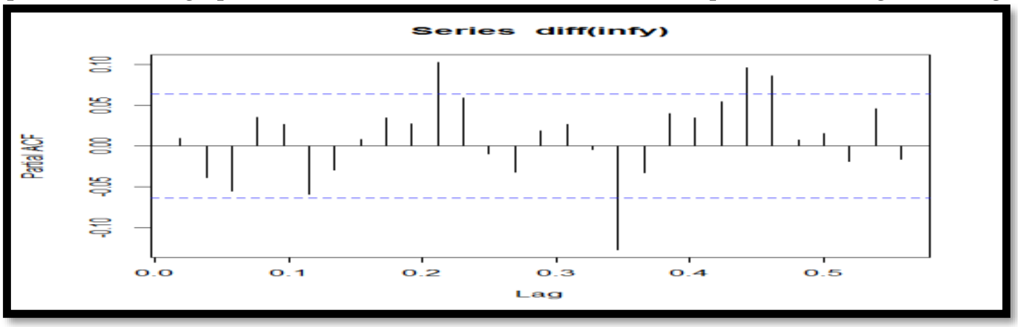


Fig. 21. Figure representing the PACF graph of the INFOSYS timeseries data

It is evident from the PACF graph in Fig. 21 , that from the initial lag (lag=1), the graph cuts off and thus decreases the correlation factor from the beginning . So p=0 is deduced from the PACF graph.

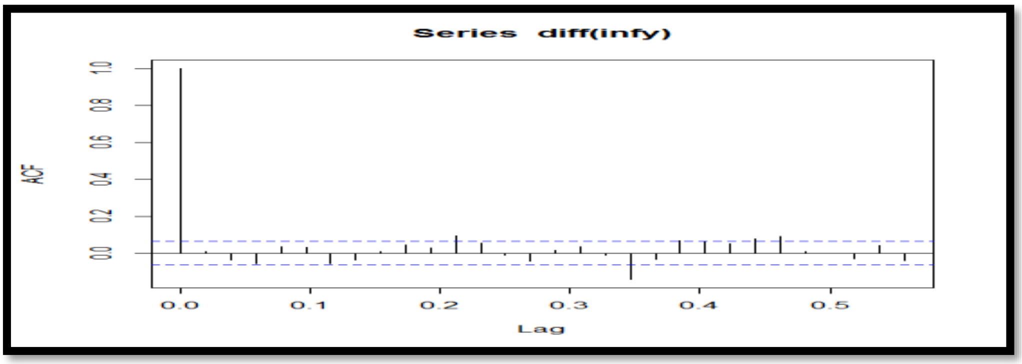


Fig. 22. Figure representing the ACF graph of the INFOSYS timeseries data

It is evident from the ACF graph in Fig.22 that from the initial lag (lag=1), the graph cuts off and thus decreases the correlation factor from the beginning. So q=0 is deduced from the ACF graph.

Thus using this method the function is ARIMA (0,1,0) for the INFOSYS timeseries data.

Using the 3 methodologies for choosing the optimal ARIMA Model of the TCS and INFOSYS timeseries data, we have obtained 2 models for each of the datasets. It is summarized in Table 8 –

	ARIMA Model using Auto-Arima	ARIMA Model using iterative minimum AIC value	ARIMA Model using PACF and ACF graphs
TCS Data	(0,1,2)	(2,1,2)	
INFOSYS Data	(3,2,1)	(0,1,0)	

Table 8. Iterative Test Results for determining value of p and q using AIC of INFOSYS Data

So , in case of TCS data, 2 predictive models are built - ARIMA(0,1,2) and ARIMA(2,1,2) using which future TCS stock closing index values can be predicted . In this paper both these models are used for prediction of TCS stock closing index values from the 4th week of March, 2021 till the 3rd week of March, 2026 (five years or 260 weeks). Two separate data reuslts have been obtained and both have been plotted in Fig. 23, Fig. 24, Fig. 25 and Fig. 26.

```
> fit2 <- Arima(tcs,order=c(0,1,2), include.drift=TRUE)
> pred <- forecast(fit2,h=260)
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2021.231       3065.658 3017.392 3113.924 2991.842 3139.475
2021.250       3067.543 2996.570 3138.517 2958.999 3176.088
2021.269       3070.712 2985.578 3155.847 2940.510 3200.914
2021.288       3073.881 2976.626 3171.136 2925.142 3222.620
2021.308       3077.050 2969.026 3185.074 2911.841 3242.259
2021.327       3080.219 2962.406 3198.032 2900.039 3260.398
2021.346       3083.388 2956.539 3210.236 2889.389 3277.386
2021.365       3086.556 2951.274 3221.839 2879.660 3293.453
2021.385       3089.725 2946.505 3232.945 2870.689 3308.761
2021.404       3092.894 2942.154 3243.635 2862.357 3323.432
2021.423       3096.063 2938.160 3253.966 2854.571 3337.555
2021.442       3099.232 2934.477 3263.986 2847.262 3351.202
2021.462       3102.401 2931.068 3273.733 2840.371 3364.431
2021.481       3105.570 2927.903 3283.236 2833.852 3377.287
2021.500       3108.738 2924.956 3292.521 2827.667 3389.810
2021.519       3111.907 2922.205 3301.609 2821.783 3402.031
2021.538       3115.076 2919.634 3310.518 2816.174 3413.978
2021.558       3118.245 2917.227 3319.263 2810.815 3425.675
2021.577       3121.414 2914.971 3327.857 2805.686 3437.142
2021.596       3124.583 2912.853 3336.312 2800.770 3448.395
```

Fig. 23. Portion of the predicted future TCS data starting from the 4th week of March, 2021 using ARIMA(0,1,2)

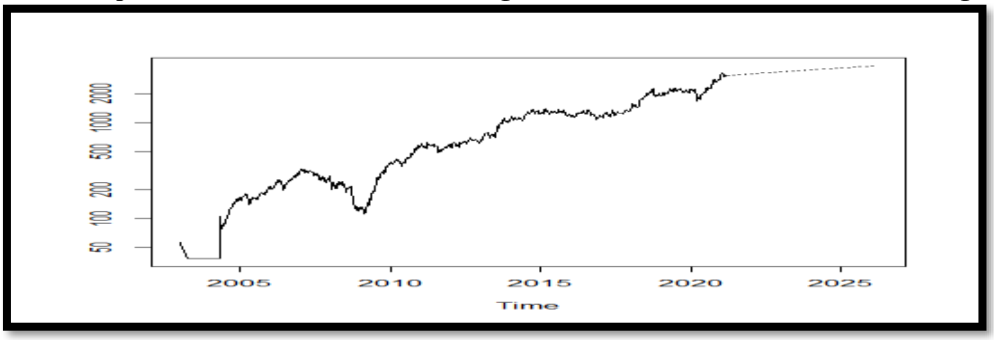


Fig. 24. Graph plotting the actual TCS data from 1st week of January, 2003 till the 3rd week of March, 2021and future predicted data from 4th week of March, 2021 till the 3rd week of March, 2026 using ARIMA(0,1,2) . Solid lines denote actual data and dotted lines denote predicted data .

```
> fit2 <- Arima(tcs,order=c(2,1,2), include.drift=TRUE)
> pred <- forecast(fit2,h=260)
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2021.231       3068.540 3020.349 3116.731 2994.838 3142.242
2021.250       3066.837 2996.358 3137.317 2959.048 3174.627
2021.269       3073.895 2989.826 3157.964 2945.323 3202.468
2021.288       3075.460 2979.339 3171.581 2928.455 3222.465
2021.308       3080.088 2974.222 3185.955 2918.179 3241.998
2021.327       3082.758 2967.933 3197.582 2907.149 3258.367
2021.346       3086.495 2963.689 3209.301 2898.679 3274.310
2021.365       3089.530 2959.249 3219.812 2890.282 3288.779
2021.385       3092.936 2955.706 3230.166 2883.060 3302.812
2021.404       3096.089 2952.263 3239.916 2876.126 3316.053
2021.423       3099.370 2949.285 3249.456 2869.834 3328.906
2021.442       3102.560 2946.477 3258.643 2863.852 3341.268
2021.462       3105.793 2943.954 3267.632 2858.282 3353.304
2021.481       3108.992 2941.601 3276.384 2852.990 3364.995
2021.500       3112.207 2939.450 3284.964 2847.998 3376.416
2021.519       3115.409 2937.450 3293.367 2843.245 3387.572
2021.538       3118.615 2935.607 3301.623 2838.729 3398.502
2021.558       3121.817 2933.897 3309.738 2834.417 3409.217
2021.577       3125.021 2932.314 3317.728 2830.302 3419.740
2021.596       3128.223 2930.847 3325.599 2826.362 3430.083
```

Fig. 25. Portion of the predicted future TCS data starting from the 4th week of March, 2021 using ARIMA(2,1,2)

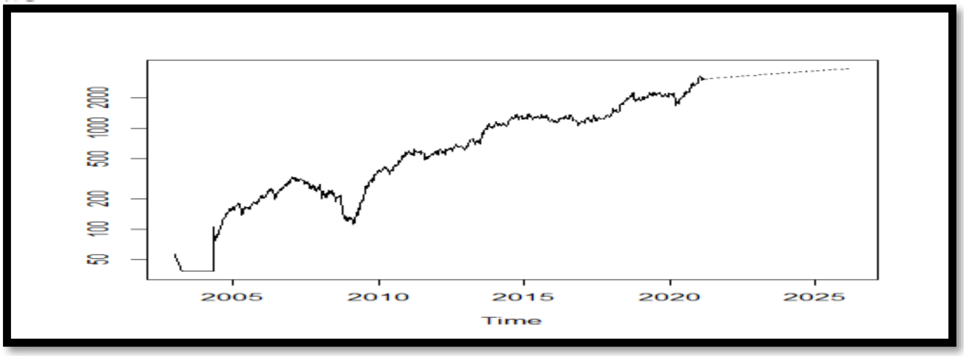


Fig. 26. Graph plotting the actual TCS data from 1st week of January, 2003 till the 3rd week of March, 2021and future predicted data from 4th week of March, 2021 till the 3rd week of March, 2026 using ARIMA(2,1,2) . Solid lines denote actual data and dotted lines denote predicted data .

In case of INFOSYS data, 2 predictive models are built - ARIMA(3,2,1) and ARIMA(0,1,0) using which future INFOSYS stock closing index values can be predicted . In this paper both these models are used for prediction of INFOSYS stock closing index values from the 4th week of March, 2021 till the 3rd week of March, 2026 (five years or 260 weeks). Two separate data reuslts have been obtained and both have been plotted in Fig. 27, Fig. 28, Fig. 29 and Fig. 30.

```
> fit2 <- Arima(infy,order=c(3,2,1), include.drift=TRUE)
Warning message:
In Arima(infy, order = c(3, 2, 1), include.drift = TRUE) :
  No drift term fitted as the order of difference is 2 or more.
> pred <- forecast(fit2,h=260)
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2021.231       1351.704 1328.917 1374.492 1316.855 1386.554
2021.250       1361.277 1328.912 1393.642 1311.779 1410.774
2021.269       1367.923 1328.717 1407.129 1307.962 1427.884
2021.288       1373.995 1329.576 1418.413 1306.062 1441.927
2021.308       1379.999 1330.785 1429.214 1304.733 1455.266
2021.327       1386.228 1332.488 1439.969 1304.039 1468.417
2021.346       1392.498 1334.454 1450.542 1303.728 1481.268
2021.365       1398.761 1336.628 1460.894 1303.736 1493.786
2021.385       1405.007 1338.958 1471.056 1303.994 1506.020
2021.404       1411.250 1341.427 1481.074 1304.465 1518.036
2021.423       1417.495 1344.014 1490.977 1305.115 1529.876
2021.442       1423.742 1346.700 1500.783 1305.917 1541.566
2021.462       1429.988 1349.473 1510.503 1306.851 1553.125
2021.481       1436.234 1352.319 1520.149 1307.897 1564.571
2021.500       1442.480 1355.230 1529.730 1309.042 1575.918
2021.519       1448.726 1358.197 1539.254 1310.275 1587.177
2021.538       1454.972 1361.215 1548.729 1311.584 1598.360
```

Fig. 27. Portion of the predicted future INFOSYS data starting from the 4th week of March, 2021 using ARIMA(3,2,1)

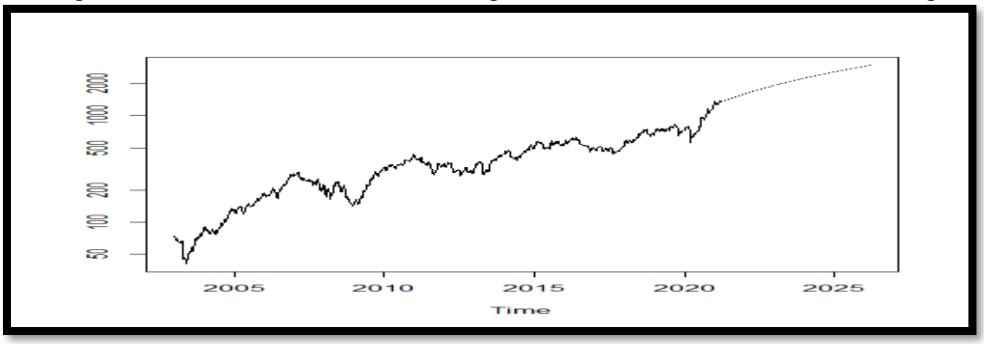


Fig. 28. Graph plotting the actual INFOSYS data from 1st week of January, 2003 till the 3rd week of March, 2021and future predicted data from 4th week of March, 2021 till the 3rd week of March, 2026 using ARIMA(3,2,1) . Solid lines denote actual data and dotted lines denote predicted data .

```
> fit2 <- Arima(infy,order=c(0,1,0), include.drift=TRUE)
> pred <- forecast(fit2,h=260)
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2021.231       1346.692 1323.891 1369.493 1311.821 1381.564
2021.250       1348.034 1315.788 1380.280 1298.718 1397.350
2021.269       1349.376 1309.883 1388.869 1288.977 1409.776
2021.288       1350.719 1305.116 1396.321 1280.975 1420.462
2021.308       1352.061 1301.076 1403.046 1274.086 1430.036
2021.327       1353.403 1297.551 1409.254 1267.985 1438.820
2021.346       1354.745 1294.418 1415.072 1262.484 1447.007
2021.365       1356.087 1291.595 1420.579 1257.456 1454.719
2021.385       1357.429 1289.025 1425.833 1252.815 1462.044
2021.404       1358.771 1286.667 1430.876 1248.498 1469.045
2021.423       1360.114 1284.490 1435.737 1244.458 1475.770
2021.442       1361.456 1282.470 1440.442 1240.657 1482.254
2021.462       1362.798 1280.587 1445.009 1237.067 1488.529
2021.481       1364.140 1278.825 1449.455 1233.663 1494.618
2021.500       1365.482 1277.173 1453.791 1230.425 1500.539
2021.519       1366.824 1275.619 1458.030 1227.338 1506.311
2021.538       1368.167 1274.154 1462.179 1224.387 1511.946
2021.558       1369.509 1272.771 1466.246 1221.561 1517.456
2021.577       1370.851 1271.462 1470.239 1218.849 1522.852
2021.596       1372.193 1270.223 1474.163 1216.243 1528.143
2021.615       1373.535 1269.047 1478.024 1213.734 1533.337
```

Fig. 29. Portion of the predicted future INFOSYS data starting from the 4th week of March, 2021 using ARIMA(0,1,0)

ISSN: 2394-7659
IMPACT FACTOR- 3.775

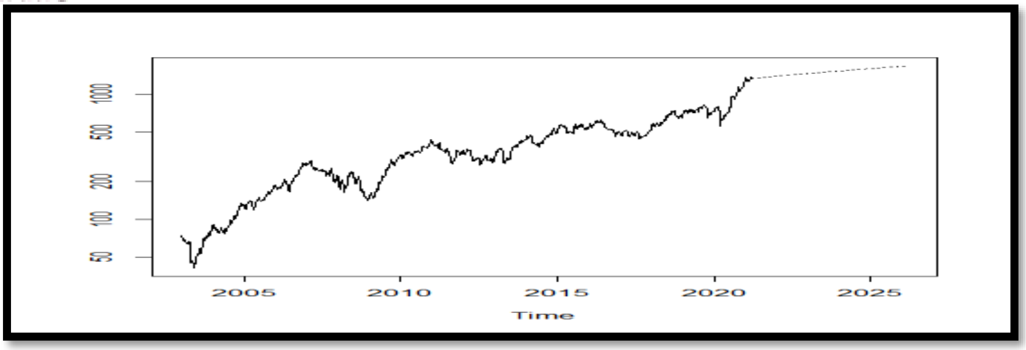


Fig. 30. Graph plotting the actual INFOSYS data from 1st week of January, 2003 till the 3rd week of March, 2021and future predicted data from 4th week of March, 2021 till the 3rd week of March, 2026 using ARIMA(0,1,0) . Solid lines denote actual data and dotted lines denote predicted data .

For both the TCS and Infosys data, 2 ARIMA predictive models are generated. Now these models are subjected to accuracy testing. The total dataset (1st week of January, 2003 till the 3rd week of March, 2021) of TCS and Infosys data is divided into Training dataset and Testing dataset. The split is done in an 80:20 ratio in favour of Training dataset over Testing dataset. The Training dataset will span from 1st week of January, 2003 – 3rd week of March, 2019 and the Testing dataset will span from 4th week of March, 2019 – 3rd week of March, 2021. For the TCS dataset, the two predictive models – ARIMA (2, 1, 2) and ARIMA (0, 1, 2) are trained using the Training dataset. Then they are used to predict the TCS weekly closing index prices from 4th week of March, 2019 – 3rd week of March, 2021. The results produced are as follows, given in Fig. 31, Fig. 32 and Fig. 33–

	ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (2,1,2)	PREDICTED VALUE WITH ARIMA (0,1,2)		ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (2,1,2)	PREDICTED VALUE WITH ARIMA (0,1,2)		ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (2,1,2)	PREDICTED VALUE WITH ARIMA (0,1,2)		ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (2,1,2)	PREDICTED VALUE WITH ARIMA (0,1,2)
27-Mar-19	2079.30	2086.97	2094.26	25-Sep-19	2059.95	2240.96	2271.17	1-Apr-20	1775.20	2361.46	2403.51	14-Oct-20	2739.00	2473.07	2524.22
3-Apr-19	2091.50	2099.41	2095.43	2-Oct-19	2047.70	2245.81	2276.56	8-Apr-20	1759.25	2365.62	2408.03	21-Oct-20	2630.15	2476.90	2528.34
10-Apr-19	2131.80	2104.52	2106.89	9-Oct-19	2037.30	2250.62	2281.90	15-Apr-20	1737.65	2369.77	2412.54	28-Oct-20	2633.60	2480.73	2532.46
17-Apr-19	2155.05	2109.30	2117.16	16-Oct-19	2051.40	2255.39	2287.18	22-Apr-20	1859.05	2373.90	2417.03	4-Nov-20	2649.60	2484.54	2536.56
24-Apr-19	2260.35	2115.25	2126.62	23-Oct-19	2194.85	2260.12	2292.42	29-Apr-20	1932.75	2378.01	2421.50	11-Nov-20	2666.05	2488.35	2540.65
1-May-19	2151.95	2122.03	2135.47	30-Oct-19	2201.85	2264.82	2297.62	6-May-20	1949.50	2382.11	2425.95	18-Nov-20	2722.05	2492.15	2544.74
8-May-19	2092.35	2129.05	2143.84	6-Nov-19	2100.95	2269.48	2302.77	13-May-20	1948.65	2386.20	2430.38	25-Nov-20	2726.80	2495.94	2548.81
15-May-19	2109.75	2135.94	2151.84	13-Nov-19	2108.80	2274.12	2307.88	20-May-20	1943.00	2390.27	2434.80	2-Dec-20	2797.30	2499.72	2552.87
22-May-19	2073.75	2142.57	2159.51	20-Nov-19	2046.65	2278.71	2312.95	27-May-20	2047.15	2394.33	2439.20	9-Dec-20	2761.55	2503.50	2556.93
29-May-19	2183.10	2148.97	2166.92	27-Nov-19	2051.00	2283.28	2317.99	3-Jun-20	2072.05	2398.37	2443.58	16-Dec-20	2872.50	2507.26	2560.97
5-Jun-19	2252.80	2155.15	2174.10	4-Dec-19	2012.85	2287.82	2322.99	10-Jun-20	2045.80	2402.41	2447.95	30-Dec-20	3093.00	2511.02	2565.01
12-Jun-19	2250.85	2161.18	2181.07	11-Dec-19	2164.95	2292.34	2327.95	17-Jun-20	2035.30	2406.43	2452.30	6-Jan-21	3174.85	2514.78	2569.03
19-Jun-19	2267.80	2167.08	2187.86	18-Dec-19	2215.60	2296.82	2332.88	24-Jun-20	2082.15	2410.43	2456.64	13-Jan-21	3260.70	2518.52	2573.05
26-Jun-19	2252.10	2172.86	2194.50	25-Dec-19	2161.70	2301.28	2337.77	1-Jul-20	2269.90	2414.43	2460.96	20-Jan-21	3291.30	2522.26	2577.06
3-Jul-19	2133.35	2178.54	2200.99	1-Jan-20	2205.85	2305.72	2342.64	8-Jul-20	2171.95	2418.41	2465.27	27-Jan-21	3203.45	2525.99	2581.06
10-Jul-19	2106.00	2184.13	2207.35	8-Jan-20	2206.90	2310.13	2347.48	15-Jul-20	2225.05	2422.38	2469.56	3-Feb-21	3176.90	2529.71	2585.05
17-Jul-19	2112.45	2189.62	2213.59	15-Jan-20	2171.05	2314.52	2352.28	22-Jul-20	2309.75	2426.34	2473.84	10-Feb-21	3108.80	2533.43	2589.04
24-Jul-19	2179.15	2195.04	2219.72	22-Jan-20	2183.75	2318.89	2357.06	29-Jul-20	2249.70	2430.29	2478.11	17-Feb-21	2980.20	2537.14	2593.01
31-Jul-19	2214.90	2200.39	2225.75	29-Jan-20	2107.75	2323.23	2361.82	5-Aug-20	2279.90	2434.23	2482.36	24-Feb-21	3006.35	2540.84	2596.98
7-Aug-19	2199.45	2205.66	2231.69	5-Feb-20	2153.40	2327.55	2366.54	12-Aug-20	2269.75	2438.16	2486.60	3-Mar-21	3050.95	2544.54	2600.94
14-Aug-19	2186.75	2210.87	2237.55	12-Feb-20	2215.75	2331.86	2371.24	19-Aug-20	2242.65	2442.08	2490.83	10-Mar-21	3110.05	2548.23	2604.89
21-Aug-19	2236.50	2216.02	2243.33	19-Feb-20	2156.15	2336.14	2375.92	26-Aug-20	2246.35	2445.99	2495.04	17-Mar-21	3050.20	2551.91	2608.83
28-Aug-19	2251.60	2221.11	2249.03	26-Feb-20	2036.20	2340.41	2380.57	2-Sep-20	2348.20	2449.88	2499.25	19-Mar-21	3054.80	2555.59	2612.77
4-Sep-19	2182.85	2226.15	2254.66	4-Mar-20	1972.35	2344.65	2385.20	9-Sep-20	2491.40	2453.77	2503.44				
11-Sep-19	2122.65	2231.13	2260.22	11-Mar-20	1658.00	2348.88	2389.81	16-Sep-20	2522.95	2457.65	2507.62				
18-Sep-19	2044.70	2236.07	2265.73	18-Mar-20	1703.15	2353.09	2394.40	23-Sep-20	2488.40	2461.52	2511.78				
				25-Mar-20	1826.10	2357.28	2398.96	30-Sep-20	2714.30	2465.38	2515.94				
								7-Oct-20	2826.55	2469.23	2520.08				

Fig. 31. Comparison of the predicted values generated by the 2 ARIMA models in contrast with the actual data value present in the Testing dataset for the TCS data

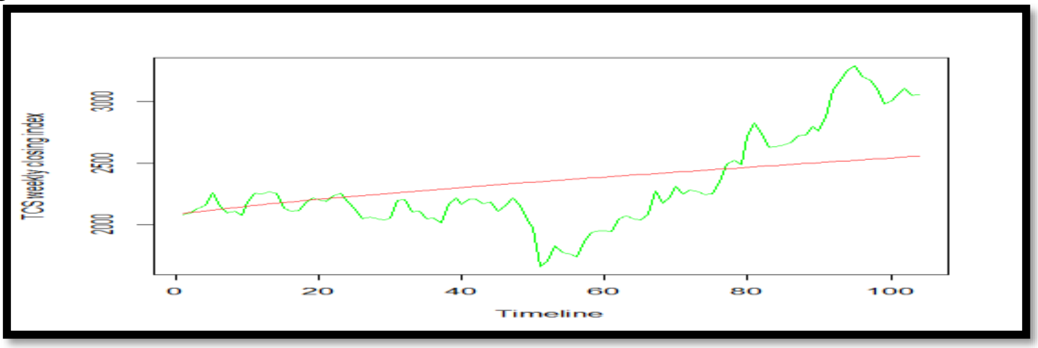


Fig. 32. Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (2,1,2) for the TCS dataset

IJERMS

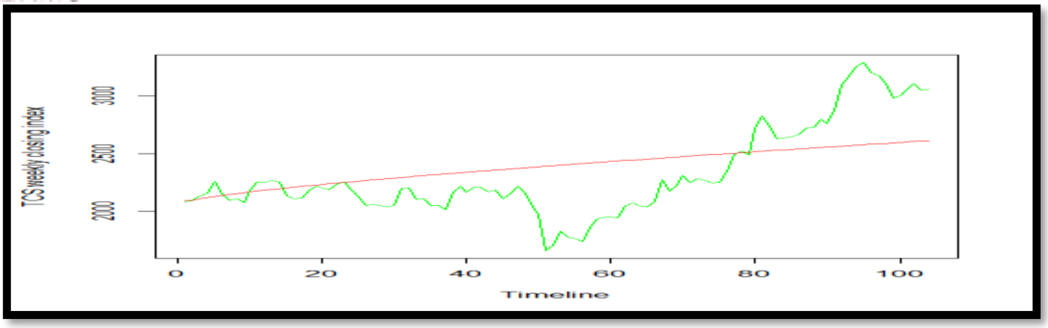


Fig. 33. Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (0,1,2) for the TCS dataset

For the INFOSYS dataset, the two predictive models – ARIMA (0, 1, 0) and ARIMA (3, 2, 1) are trained using the Training dataset. Then they are used to predict the INFOSYS weekly closing index prices from 4th week of March, 2019 – 3rd week of March, 2021. The results produced are as follows, given in Fig. 34, Fig. 35 and Fig. 36–

	ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (0,1,0)	PREDICTED VALUE WITH ARIMA (3,2,1)
27-Mar-19	757.10	758.99	759.58
3-Apr-19	759.85	766.62	766.10
10-Apr-19	723.85	772.65	771.20
17-Apr-19	727.85	777.87	775.82
24-Apr-19	749.95	782.56	780.04
1-May-19	723.55	786.87	783.95
8-May-19	713.85	790.91	787.63
15-May-19	709.20	794.72	791.11
22-May-19	726.65	798.34	794.45
29-May-19	734.70	801.81	797.66
5-Jun-19	754.85	805.15	800.76
12-Jun-19	750.65	808.38	803.76
19-Jun-19	748.05	811.51	806.68
26-Jun-19	739.90	814.55	809.52
3-Jul-19	715.40	817.51	812.30
10-Jul-19	784.95	820.40	815.02
17-Jul-19	789.90	823.23	817.68
24-Jul-19	792.80	826.00	820.29
31-Jul-19	774.65	828.71	822.86
7-Aug-19	764.10	831.37	825.38
14-Aug-19	792.90	833.99	827.87
21-Aug-19	785.00	836.56	830.31
28-Aug-19	814.30	839.10	832.73
4-Sep-19	829.20	841.59	835.11
11-Sep-19	831.05	844.06	837.46
18-Sep-19	794.05	846.48	839.79

	ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (0,1,0)	PREDICTED VALUE WITH ARIMA (3,2,1)
25-Sep-19	793.55	848.88	842.09
2-Oct-19	789.55	851.25	844.36
9-Oct-19	767.80	853.59	846.61
16-Oct-19	643.30	855.90	848.84
23-Oct-19	650.60	858.19	851.05
30-Oct-19	695.80	860.46	853.24
6-Nov-19	704.50	862.70	855.40
13-Nov-19	712.80	864.92	857.56
20-Nov-19	690.70	867.12	859.69
27-Nov-19	698.35	869.30	861.80
4-Dec-19	714.25	871.46	863.90
11-Dec-19	729.30	873.60	865.99
18-Dec-19	733.25	875.73	868.06
25-Dec-19	731.75	877.84	870.12
1-Jan-20	727.75	879.93	872.16
8-Jan-20	775.35	882.00	874.19
15-Jan-20	761.90	884.06	876.20
22-Jan-20	777.90	886.11	878.21
29-Jan-20	787.85	888.14	880.20
5-Feb-20	773.80	890.16	882.18
12-Feb-20	797.45	892.16	884.16
19-Feb-20	798.35	894.16	886.12
26-Feb-20	746.30	896.13	888.07
4-Mar-20	739.25	898.10	890.01
11-Mar-20	555.60	900.06	891.94
18-Mar-20	593.55	902.00	893.86
25-Mar-20	640.30	903.94	895.77

	ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (0,1,0)	PREDICTED VALUE WITH ARIMA (3,2,1)
1-Apr-20	641.10	905.86	897.68
8-Apr-20	637.10	907.77	899.57
15-Apr-20	633.05	909.67	901.46
22-Apr-20	661.05	911.57	903.34
29-Apr-20	674.25	913.45	905.21
6-May-20	687.75	915.32	907.08
13-May-20	668.95	917.19	908.93
20-May-20	680.15	919.04	910.78
27-May-20	708.05	920.89	912.63
3-Jun-20	717.75	922.73	914.46
10-Jun-20	700.95	924.56	916.29
17-Jun-20	720.45	926.38	918.12
24-Jun-20	735.90	928.20	919.93
1-Jul-20	794.40	930.00	921.74
8-Jul-20	783.20	931.80	923.55
15-Jul-20	936.70	933.60	925.35
22-Jul-20	962.75	935.38	927.14
29-Jul-20	950.10	937.16	928.93
5-Aug-20	948.20	938.93	930.71
12-Aug-20	967.15	940.69	932.49
19-Aug-20	938.35	942.45	934.26
26-Aug-20	914.30	944.20	936.02
2-Sep-20	940.05	945.94	937.79
9-Sep-20	981.90	947.68	939.54
16-Sep-20	1007.20	949.42	941.29
23-Sep-20	1010.25	951.14	943.04
30-Sep-20	1055.75	952.86	944.78
7-Oct-20	1158.00	954.58	946.52

	ACTUAL VALUE	PREDICTED VALUE WITH ARIMA (0,1,0)	PREDICTED VALUE WITH ARIMA (3,2,1)
14-Oct-20	1137.05	956.29	948.25
21-Oct-20	1092.15	957.99	949.98
28-Oct-20	1061.95	959.69	951.71
4-Nov-20	1090.95	961.38	953.43
11-Nov-20	1123.85	963.07	955.15
18-Nov-20	1139.15	964.75	956.86
25-Nov-20	1137.25	966.43	958.57
2-Dec-20	1153.85	968.10	960.27
9-Dec-20	1154.40	969.77	961.97
16-Dec-20	1220.75	971.43	963.67
30-Dec-20	1293.50	973.09	965.36
6-Jan-21	1371.85	974.74	967.05
13-Jan-21	1317.05	976.39	968.74
20-Jan-21	1322.60	978.03	970.42
27-Jan-21	1270.80	979.67	972.10
3-Feb-21	1305.00	981.30	973.78
10-Feb-21	1290.10	982.94	975.45
17-Feb-21	1266.25	984.56	977.12
24-Feb-21	1304.05	986.18	978.79
3-Mar-21	1345.70	987.80	980.45
10-Mar-21	1384.25	989.42	982.12
17-Mar-21	1345.35	991.03	983.77
19-Mar-21	1345.35	992.63	985.43

Fig. 34. Comparison of the predicted values generated by the 2 ARIMA models in contrast with the actual data value present in the Testing dataset for the INFOSYS data

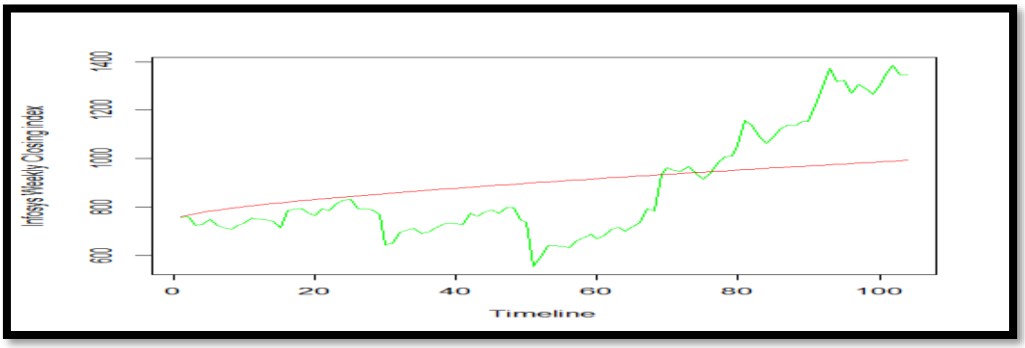


Fig. 35. Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (0,1,0) for the INFOSYS dataset

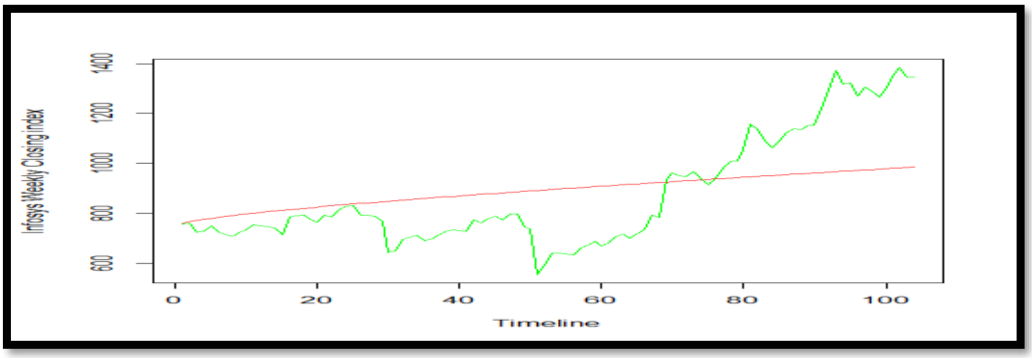


Fig. 36. Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (3,2,1) for the INFOSYS dataset

In order to the perform accuracy testing, the following error metrics were calculated – MAPE (Mean Absolute Percentage Error) and MPE (Mean Percentage Error). The error metrics observed have been tabulated in Table 9 (for the TCS dataset) and Table 10 (for the INFOSYS dataset).

ACCURACY TEST USING ARIMA (2,1,2)		ACCURACY TEST USING ARIMA (0,1,2)	
MAPE (%)	**MPE (%)**	**MAPE (%)**	**MPE (%)**
10.70464	3.276394	11.22923	4.946187

Table 9. Accuracy testing results of TCS Data

ACCURACY TEST USING ARIMA (0,1,0)		ACCURACY TEST USING ARIMA (3,2,1)	
MAPE (%)	**MPE (%)**	**MAPE (%)**	**MPE (%)**
17.24371	7.071973	16.85683	6.237845

Table 10. Accuracy testing results of INFOSYS Data

8. CONCLUSION AND SCOPE OF FUTURE IMPROVEMENT

The research paper has tried to study, analyse the TCS and Infosys weekly stock prices data from 1st week of January, 2003 till the 3rd week of March, 2021 (948 weeks), and generate a predictive model for forecasting the future stock prices of TCS and Infosys. The paper has performed a accuracy testing to check the magintude of success of the prediction done by the generated predictive models. It has achieved a MAPE and MPE rate which is < 12% in 75% of the tested ARIMA models. This paper might help future researchers in investigating further in the field of Timeseries analysis and generation of predictive models.

Artificial neural networks (ANN) are being used widely now to predict the stock markets. Facebook's open source library Prophet can be used for analysing and predicting timeseries data.

REFERENCES

1. Mohankumari, C.; Vishukumar M and Nagaraja Rao Chillale, "Analysis of Daily Stock Trend Prediction Using Arima Model", International Journal of Mechanical Engineering and Technology (IJMET), Volume 10, Issue No. 01, (January 2019), pp. 1772-1792.
2. B. Uma Devi, D.Sundar and Dr. P. Alli, "An Effective Time Series Analysis for Stock Trend Prediction Using ARIMA Model for Nifty Midcap-50", International Journal of Data Mining & Knowledge Management Process (IJDKP), Volume 3, Issue No. 1, (January 2013), pp. 65-78.
3. Nazmul Islam, "Forecasting Bangladesh's Inflation through Econometric Models", American Journal of Economics and Business Administration, Volume 9, Issue No. 3, (2017), pp. 56-60.
4. Edward A. and Manoj J., "Forecast Model Using Arima for Stock Prices of Automobile Sector", International Journal of Research in Finance and Marketing, Volume 6, Issue No. 4, (April 2016), pp. 1-9.
5. Mohamed Ashik. A. and Senthamarai Kannan. K, "Forecasting National Stock Price using Arima Model", Global and Stochastic Analysis , Volume 3, Issue No. 1 , (January 2017), pp. 77-81.
6. Mahantesh C. Angadi and Amogh P. Kulkarni, "Time Series Data Analysis for Stock Market Prediction using Data Mining Techniques with R", International Journal of Advanced Research in Computer Science ,Volume 6, Issue No. 6 ,(July-August 2015), pp. 104-108.
7. Kamalakannan J , Indrani Sengupta and Snehaa Chaudhury, "Stock Market Prediction using Time Series Analysis" , Computing Communications and Data Engineering Series , Volume 01, Issue No. 03, (2018), pp. 1-5.
8. Ayodele A. Adebiyi, Aderemi O. Adewumi and Charles K. Ayo, "Stock Price Prediction Using The ARIMA Model", Proceedings of the 16th IEEE International Conference on Computer Modelling and Simulation (UKSim-AMSS), (March 2014), pp. 105 -111.
9. Nitin Merh, Vinod P. Saxena and Kamal Raj Pardasani, "A Comparison Between Hybrid Approaches of ANN And Arima For Indian Stock Trend Forecasting", Business Intelligence Journal , Volume 3, Issue No. 2, (July 2010), pp.23-43.