IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

BSE Sensex Closing Index Data Analysis and Forecasting using the ARIMA Model

Debaditya Raychaudhuri

Dept. of Computer Science , Chandernagore College , Govt. of West Bengal, B.D. – 413 , Sector -1 , Salt Lake City , Kolkata – 700064

Corresponding Author: rana22021989@gmail.com, Phone: 9830182558 / 8777023447

DOI: https://doi.org/10.26438/ijcse/v7i6.379389 | Available online at: www.ijcseonline.org

Accepted: 13/Jun/2019, Published: 30/Jun/2019

Abstract **-** The Bombay Stock Exchange (BSE) is India's premier and most prestigious stock market. A stock market is a facilitation centre for trading (buying and selling) of stocks of various companies. Its index is calculated as a combination of stock prices of several companies enlisted in the exchange. The stock market is characterized by its endless and unpredictable troughs and crests. This paper attempts to analyse the BSE Sensex Closing Index data over a span of the last decade collected on the last day of the month (from May, 2009 to April, 2019). It also attempts to predict the BSE Sensex closing data for a future span of 10 years at a monthly frequency. The paper also does a accuracy testing of the predictive model generated. This work would be beneficial for both the nation and a trading individual. The Stock market indices reflect the health of a nation's economy and its direction and growth. A trading individual would benefit in his pursuit of profit making by taking correct investment decisions based on accurate predictions made. The paper uses the ARIMA model for timeseries analysis and for generating a predictive model for making future forecasts.

Keywords**:** Arima model, Sensex forecasting, Short-term prediction, Stock market prediction, Time Series analysis.

I. INTRODUCTION

The Bombay Stock Exchange (BSE) is India's oldest stock exchange which began its prestigious journey back in 1875. It is also Asia's first stock market.

The stock market is a centre or hub which allows traders to trade equities, shares, debentures and securities. The transactions can be done both in a physical manner as well as virtual (online) manner. The stock exchange is an integral part of a nation's economy because it is through this place that new investments come into a company. The company can grow, expand and generate revenue which in turn benefits the nation as a whole. It also signals whether there is a investment-friendly atmosphere prevalent or a sluggish bear-like situation in the country.

Forecasting the stock market is difficult, challenging and essential. An assumption needs to be made based on which stock market prediction can be done. That is the data that is available in public domain has some relationship capabilities amongst themselves which will help in predicting the future trends of the stock market.

Individual investors always search for mechanisms of profit-making in the stock market which will enable them to make low-risk investments and get maximum return of investment. This is the biggest motivation behind this kind of research work which deals with developing stock market predictive models.

Now coming to the data of the stock market, the data is not a set of random, discontinuous values. But it is a timeseries data, which holds variation of one variable against the continuous timeline collected at regular time intervals in this case, monthly). For building the stock market predictive model, in this paper I have utilized the ARIMA model of timeseries analysis. The ARIMA model is an effective, robust and efficient model for short-term market prediction. It follows an algorithm of converting a non-stationary timeseries into a stationary timeseries before using it for forecasting. ARIMA models are based on statistical techniques and methodologies.

The programming language used in this research paper is R and the IDE used is the RGUI (64 bit). R is a very popular data analysis and statistical programming language. It is being used to generate a predictive model using the ARIMA model.

The paper mainly aims at data exploratory analysis of the BSE Sensex closing index data of each month of the last 10

years , generate a predictive model for forecasting future Sensex closing data and an accuracy checking test which produces results in terms of MAPE (Mean Absolute Percentage Error) and MPE (Mean Percentage Error).

Section I of the paper deals with a brief and crisp introduction regarding the paper. Section II elaborates the objectivity of the study. Section III enlists a set of related work in this field. Section IV harps upon the literature and mathematics associated with the ARIMA model. Section V talks about the data source used in the experiment and Section VI gives a pictorial representation of the methodology. Section VII gives the results achieved in this experiment and Section VIII concludes the research work.

II. OBJECTIVE

As already stated before, this research paper aims to analyse the BSE Sensex closing index data of the last day of every month from May, 2009 to April, 2019. Then a predictive model is built using the ARIMA modelling technique to forecast the BSE Sensex closing index monthly data from May, 2019 to April, 2029. Then the dataset is divided into training and testing dataset for accuracy checking purposes.

III. RELATED WORK

There are many notable and commendable works in this field. The author D.Banerjee had developed a predictive model for forecasting Indian stock market [1]. He used the monthly stock indices data for prediction using ARIMA. The authors Ayodele A. Adebiyi , Aderemi O. Adewumi , Charles K. Ayo used ARIMA model for designing predictive models for forecasting stock prices of Nokia and Zenith Bank [2]. They used NYSE and NSE data. The authors Aloysius Edward and Jyothi Manoj had built a forecast model using ARIMA for predicting stock prices of various Automobile sector stocks like Bajaj,Tata Motors,Hero and Mahindra [3]. The authors Mohamed Ashik A, Senthamarai Kannan K had used ARIMA model for forecasting stock prices of the National Stock Exchange NIFTY 50 stocks [4]. The authors Mahantesh C. Angadi , Amogh P. Kulkarni performed time series analysis using ARIMA model for stock market forecasting [5]. The authors Prapanna Mondal, Labani Shit and Saptarsi Goswami studied the effectiveness of time series modelling using /arima for forecasting stock prices of various sectors – IT , infrastructure , banking, automobiles , power , FMCG , steel [6]. The authors J. Kamalakannan , I. Sengupta and S. Chaudhury performed stock market prediction using ARIMA model for stock prices of APPLE Inc [7].The authors J.V.N. Lakshmi, Ananthi Sheshasaayee performed a data analysis of temperature dataset using Hadoop techniques [8]. The authors Himanshi , Komal Kumar Bhatia used KNN classifier to analyse undergraduate students' salary[9].

IV. LITERATURE RELATED TO ARIMA MODELLING TECHNIQUE

The *ARIMA (Auto-Regressive Integrated Moving Average) Model* is generally used for analysis of time series data and prediction future values of the same time series. Essentially, Time Series is a sequence of numerical data obtained at regular time intervals (frequency) over a period of time. In other words it is a sequence of data a variable assumes over a period of time at fixed intervals.In an univariate time series there is a single variable, whose data changes over regular time intervals.

A normal linear regression equation should be like:

$y=mx+c$ (1).

[y: dependant variable , x: independant variable] . In a Time series, in general there is a single variable (univariate system) whose value changes over time. Here the value of the variable is dependant on its own past historic data.

Here, $y_t = y_{t-1} + Error/Randomness$ (2).

Some classical examples of Time Series data are – monthly sales data of a company, yearly GDP of a nation, daily temperature of a city, stock prices etc.

The principal components of a Time Series are – *Trend*, *Seasonality*, *Cyclicity* and *Randomness/Error*.

Trend is a general direction in which the series is moving. Trend is the increase or decrease in the series over a period of time; it persists over a long period of time. It is either a upward trend or downward trend. *Seasonality* is a regular pattern of up and down fluctuations. It generally happens over a short span of time, i.e. within a year. *Cyclicity* is medium-term variation caused by circumstances which repeat in irregular intervals. Its frequency is longer than seasonal (more than a year). *Randomness / Error* refers to variations which occur due to unpredictable patterns and do not repeat in any particular pattern. After the 1st three components are taken out of the series we are left with the randomness.

A time series can be decomposed into its components – trend, seasonality and randomness. The decomposition can be of two types : Additive decomposition and Multiplicative decomposition .

Additive decomposition:

$Y_t = Trend_t + Seasonality_t + Randomness_t$ (3)

Multiplicative decomposition:

$Y_t = Trend_t * Seasonality_t * Randomness_t$ (4)

The additive decomposition is the most appropriate if the magnitude of the seasonal and trend variations does not vary with the level of the time series. When the variation in the seasonality and trend seems to be varying with the leve of the time series , then a multiplicative decomposition is more appropriate.

In an *Auto Regression (AR) model*, we forecast the variable using a linear combination of *past values of the*

variable. The term *auto*regression indicates that it is a regression of the variable against itself. Thus, an autoregressive model of order *'p'* can be written as -

$$y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + \cdots + a_p * y_{t-p} + e_t \quad (5)$$

where $\mathbf{e_t}$ is white noise. This is like a multiple regression but with *lagged values* of y_t as predictors. We refer to this as an **AR(p) model**, an autoregressive model of order p.

AR(1) : $y_t = a_0 + a_1 * y_{t-1} + e_t$ (6) AR(2) : $y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + e_t$ (7)

So, in order to represent y(t) in a AR model of order *'p'* in terms of a linear function we can write :

$$\mathbf{Y_t = F(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, e_t)} \quad (8)$$

In a *Moving Average (MA) model* ,rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors (white noise) in a regression-like model. Thus, an moving average model of order *'q'* can be written as –

$$\mathbf{y_t = b_0 + e_t + b_1 * e_{t-1} + b_2 * e_{t-2} + \cdots + b_q * e_{t-q}} \quad (9)$$

where $\mathbf{e_t}$ is white noise. We refer to this as an **MA(q) model**, a moving average model of order q.

MA(1) : $y_t = b_0 + e_t + b_1 * e_{t-1}$ (10)

MA(2) : $y_t = b_0 + e_t + b_1 * e_{t-1} + b_2 * e_{t-2}$ (11)

So, in order to represent y_t in a MA model of order *'q'* in terms of a linear function we can write : $\mathbf{Y_t = F(e_{t-1}, e_{t-2}, \ldots, e_{t-q}, e_t)}$ (12)

When we combine the AR and the MA model then 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 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 –

$$\mathbf{y_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + .. + a_p * y_{t-p} + e_t + b_0 + b_1 * e_{t-1} + b_2 * e_{t-2} + ... + b_q * y_{t-q} + e_t} \quad (13)$$

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

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

ARMA (1, 1): $y_t = a_0 + a_1 * y_{t-1} + b_0 + b_1 * e_{t-1} + e_t$ (16)

A *stationary* time series is one whose statistical properties such as mean, variance, are all constant over time. Most statistical forecasting methods are based on the assumption that the time series can be rendered approximately stationary (i.e., "stationarized") through the use of mathematical transformations like differencing. A time series can be used for predictive purposes using the ARIMA if and only if it is a stationary series. In other words we can say that stationary series does not have a upward or downward trend. Stationarity of a Time Series can be tested 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 make a non-stationary time series stationary is to compute the differences between consecutive observations. 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, thus converts a non-stationary series into a stationary one. In other words we can say that, differencing de-trends a time series. The differenced series is the change between consecutive observations in the original series, and can be written as $y'_t = y_t - y_{t-1}$. The differenced series will have only T−1 values, since it is not possible to calculate a difference y'_1 for the first observation. The number of times differencing is done on the series is called the *"order of differencing"*. Denoted by'd'.

1st order differencing: $\mathbf{y'_t = y_t - y_{t-1}}$ (17)

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

2nd order differencing:

$\mathbf{y''_t = y'_t - y'_{t-1}}$ (18)

$y''_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$ (19)

$$\mathbf{y''_t = y_t - 2 * y_{t-1} + y_{t-2}.} \quad (20)$$

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

When we combine the AR and the MA model then we get the ARMA model. Now we combine it with the Differencing that was needed for achieving the stationarity . It is called 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 y_t 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 –

$$\mathbf{y'^{\,d}_t = a_0 + a_1 * y_{t-1} + a_2 * y_{t-2} + .. + a_p * y_{t-p} + e_t + b_0 + b_1 * e_{t-1} + b_2 * e_{t-2} + .. + b_q * y_{t-q} + e_t} \quad (21)$$

For a time series, the *Partial AutoCorrelation Function (PACF)* between x_t and x_{t-h} is defined as the conditional correlation between x_t and x_{t-h} , conditional on x_{t-h+1}, ... , x_{t-1}, the set of observations that come between the time points *t* and *t*−*h*. At lag h, this is the correlation between series values that are h intervals apart, accounting for the values of the intervals between. The order of PACF is 'p' which is needed for the AR model.

For a time series, *AutoCorrelation Function (ACF)* between x_t and x_{t-h} is defined as the correlation between x_t and x_{t-h} . At lag h, this is the correlation between the values of x_t and x_{t-h} *without taking into consideration the interval values (as was the case in PACF)* . The order of ACF is 'q' which is needed for the MA model. Using the parameters p,d and q the ARIMA model is built which will be used for predictive purposes.

V. SOURCE OF THE DATA USED FOR THE EXPERIMENT

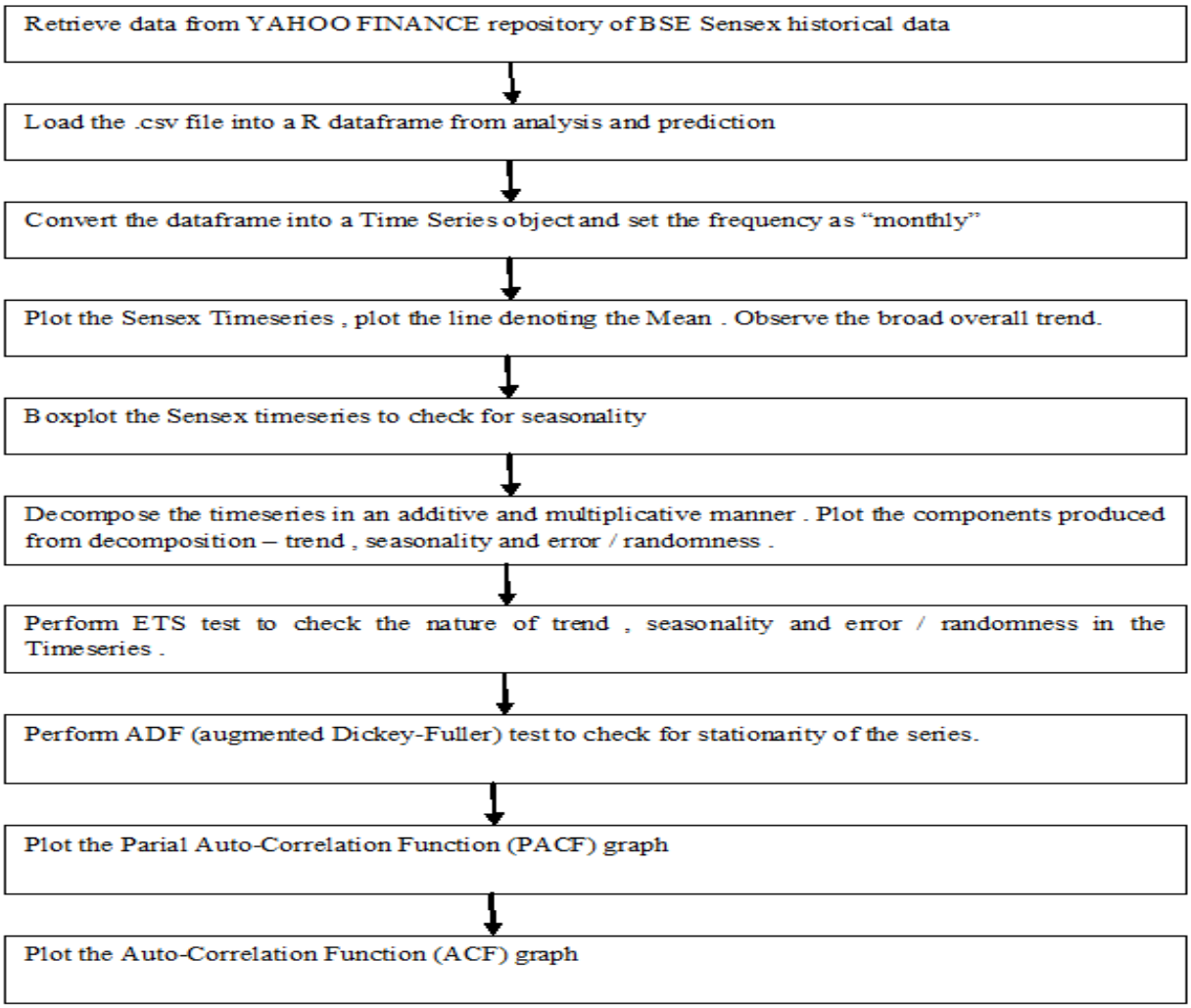
The data used for this experiment in this research paper has been retrieved as a CSV (Comma Separated Values) file from YAHOO FINANCE repository of BSE Sensex

historical data. The data is of Sensex index of the last day of each month from May, 2009 to April, 2019. The data spans over 10 years (each having 12 months), thus creating a dataset of 120 tuples (rows) . There are 7 attributes (columns) in the dataset – Date, Open, High, Low, Close, Adjusted Close and Volume. In this paper the "Closing" attribute has been used for analysis and forecasting, the other attributes (columns) has been ignored. The csv file has been converted into a dataframe in R for further processing.

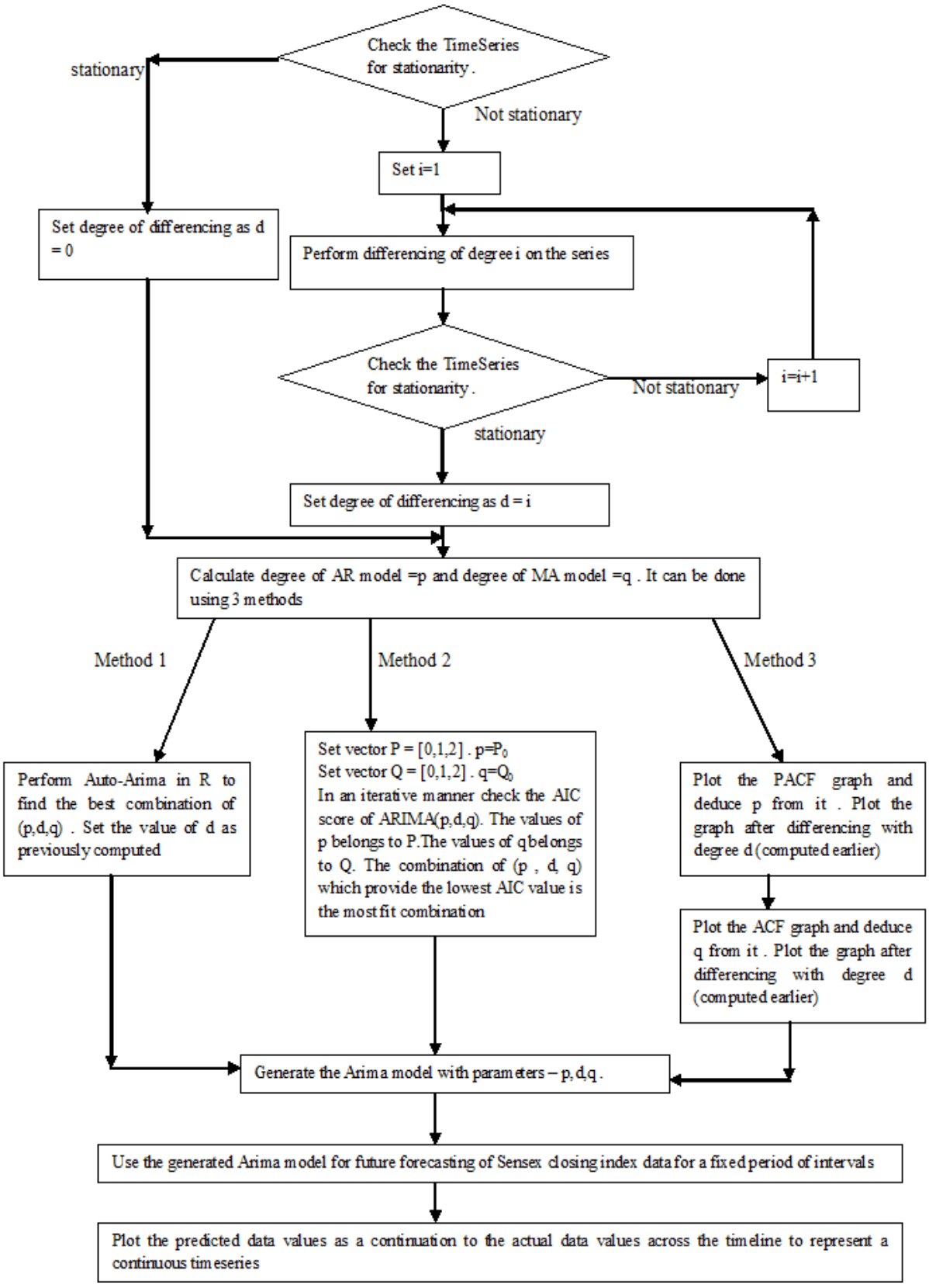
VI. METHODOLOGY OF THE EXPERIMENT

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

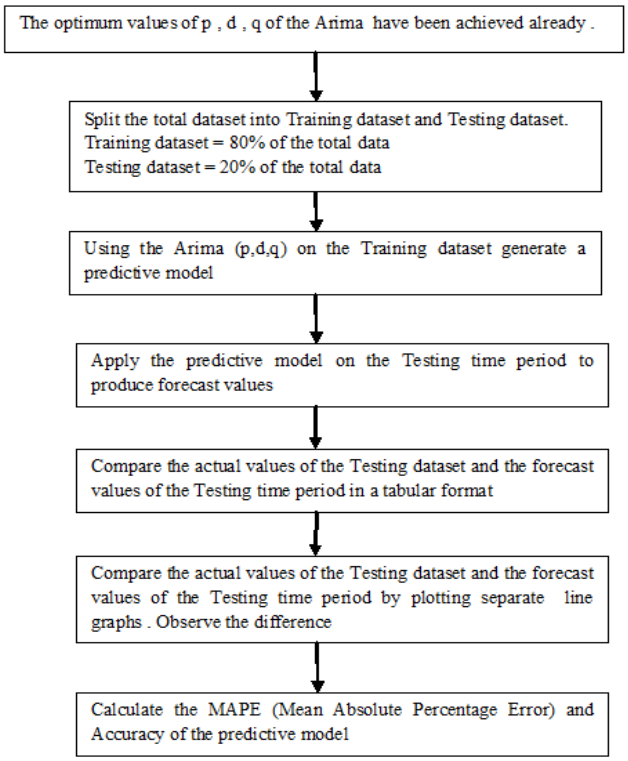
i) Data Exploratory Analysis



ii) Forecasting Using The Predictive Model

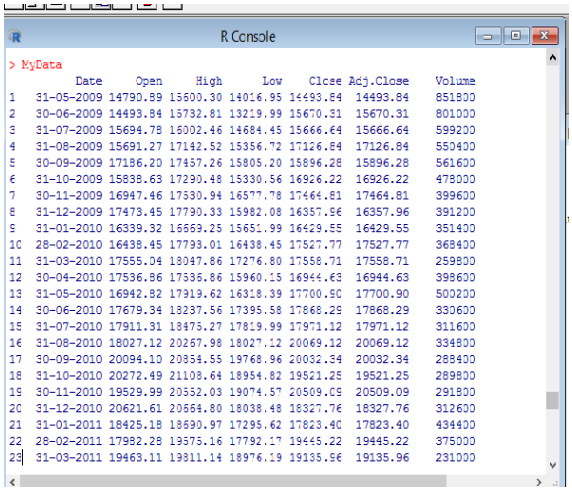


iii) Accuracy Testing Using The Trained Model



VII. RESULTS AND DISCUSSION

The experiment has been conducted using the programming language R and the IDE used is RGUI. The data is first retrieved from YAHOO FINANCE repository of BSE Sensex historical data. The data is the sensex index of the last day of each month over the span of the last 10 years, i.e. from May, 2009 till April, 2019. The snapshot of a portion of the data retrieved is given in Fig. 1 :

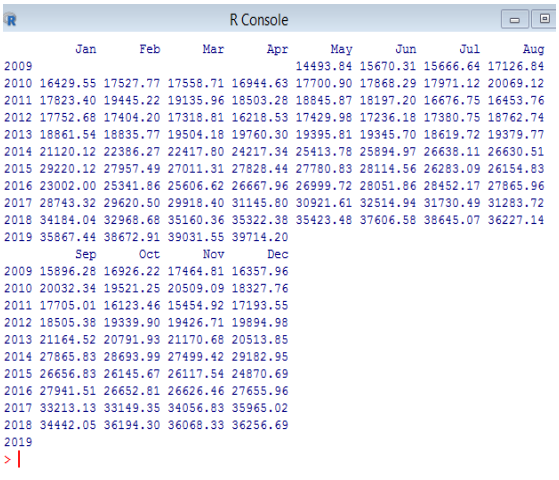

R Console

```
> MyData
         Date     Open     High      Low    Close Adj.Close    Volume
1  31-05-2009 14790.89 15600.30 14016.95 14493.84  14493.84    851800
2  30-06-2009 14493.84 15732.81 13219.99 15670.31  15670.31    801000
3  31-07-2009 15694.78 16002.46 14684.45 15666.64  15666.64    599200
4  31-08-2009 15691.27 17142.52 15356.72 17126.84  17126.84    550400
5  30-09-2009 17186.20 17437.26 15805.20 15896.28  15896.28    561600
6  31-10-2009 15838.63 17290.48 15330.56 16926.22  16926.22    478000
7  30-11-2009 16947.46 17530.94 16577.78 17464.81  17464.81    399600
8  31-12-2009 17473.45 17790.33 15982.08 16357.96  16357.96    391200
9  31-01-2010 16339.32 16563.25 15651.99 16429.55  16429.55    351400
10 28-02-2010 16438.45 17793.01 16438.45 17527.77  17527.77    368400
11 31-03-2010 17555.04 18047.86 17276.80 17558.71  17558.71    259800
12 30-04-2010 17536.86 17536.86 15960.15 16944.63  16944.63    398600
13 31-05-2010 16942.82 17919.62 16318.39 17700.90  17700.90    500200
14 30-06-2010 17679.34 18237.56 17395.58 17868.29  17868.29    330600
15 31-07-2010 17911.31 18475.27 17819.99 17971.12  17971.12    311600
16 31-08-2010 18027.12 20267.98 18027.12 20069.12  20069.12    334800
17 30-09-2010 20094.10 20854.55 19768.96 20032.34  20032.34    288400
18 31-10-2010 20272.49 21108.64 18954.82 19521.25  19521.25    289800
19 30-11-2010 19529.99 20552.03 19074.57 20509.09  20509.09    291800
20 31-12-2010 20621.61 20664.80 18038.48 18327.76  18327.76    312600
21 31-01-2011 18425.18 18690.97 17295.62 17823.40  17823.40    434400
22 28-02-2011 17982.28 19575.16 17792.17 19445.22  19445.22    375000
23 31-03-2011 19463.11 19811.14 18976.19 19135.96  19135.96    231000
```

Fig. 1: A snapshot of a portion of the whole data retrieved

For the experiment only the "Close" index is used, and the other members of the dataset are ignored. So, the data is preprocessed to achieve only the required attribute along with the Date. Then the dataset is translated into

a timeseries called Sensex(frequency being monthly) . A snapshot of the processed Timeseries dataset is given below in Fig. 2 :


R Console

```
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
2009                                    14493.84 15670.31 15666.64 17126.84
2010 16429.55 17527.77 17558.71 16944.63 17700.90 17868.29 17971.12 20069.12
2011 17823.40 19445.22 19135.96 18503.28 18845.87 18197.20 16676.75 16453.76
2012 17752.68 17404.20 17318.81 16218.53 17429.98 17236.18 17380.75 18762.74
2013 18861.54 18835.77 19504.18 19760.30 19395.81 19345.70 18619.72 19379.77
2014 21120.12 22386.27 22417.80 24217.34 25413.78 25894.97 26638.11 26630.51
2015 29220.12 27957.49 27011.31 27828.44 27780.83 28114.56 26283.09 26154.83
2016 23002.00 25341.86 25606.62 26667.96 26999.72 28051.86 28452.17 27865.96
2017 28743.32 29620.50 29918.40 31145.80 30921.61 32514.94 31730.49 31283.72
2018 34184.04 32968.68 35160.36 35322.38 35423.48 37606.58 38645.07 36227.14
2019 35867.44 38672.91 39031.55 39714.20
          Sep      Oct      Nov      Dec
2009 15896.28 16926.22 17464.81 16357.96
2010 20032.34 19521.25 20509.09 18327.76
2011 17705.01 16123.46 15454.92 17193.55
2012 18505.38 19339.90 19426.71 19894.98
2013 21164.52 20791.93 21170.68 20513.85
2014 27865.83 28693.99 27499.42 29182.95
2015 26656.83 26145.67 26117.54 24870.69
2016 27941.51 26652.81 26626.46 27655.96
2017 33213.13 33149.35 34056.83 35965.02
2018 34442.05 36194.30 36068.33 36256.69
2019
>
```

Fig. 2: A snapshot of the total Timeseries "Sensex" dataset from May, 2009 till April, 2019

Data exploratory analysis is done by plotting the timeseries graph along with its mean line to observe the trend . The graphical representation of the Sensex timeseries and its mean line is shown below. As is evident from the plotted graphs Fig. 3 and Fig. 4, there is a overall increasing or upward trend in the data.

Fig. 3: Graph representing the sensex timeseries data

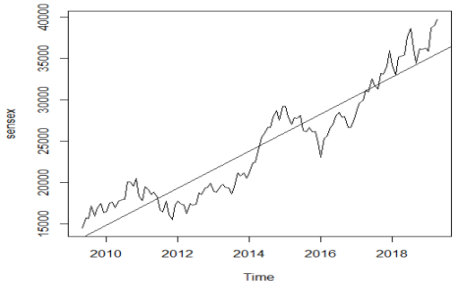


Fig. 4: Graph representing the timeseries data and mean

A boxplot of the time series data is plotted to check for seasonality. It is given as Fig. 5

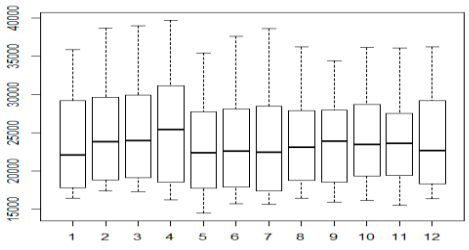


Fig. 5: Boxplot of the sensex timeseries data

The timeseries is decomposed into its principal components – Trend, Seasonality and Randomness / Error. The graphical representation the decomposed components of the Sensex data is given below in Fig. 6-

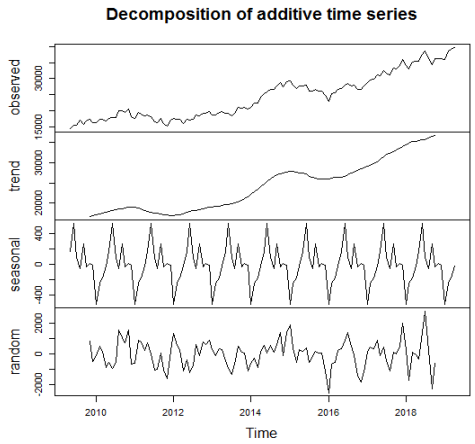


Fig 6 : Graph representing the decomposed components of the Sensex Timeseries

The ETS (Error , Trend , Seasonality) test is performed on the sensex timeseries . The results obtained are given below in Table 1 :

Table1 : ETS Test Results

Data	Error	Trend	Season-ality
Sensex	Multiplicative	Additive	None

Source : Results achieved by programming in R

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

Table2 : ADF Test Results

Data	T-Statistic	P-Value	Conclusion
Sensex	-1.6537	0.7204	Not stationary

Source : Results achieved by programming in R

As is evident from the ETS test and the ADF test , the Sensex data has a upward additive trend and the timeseries is not a stationary one . In order to build the Arima model the timeseries has to be made stationary .

Usind d=1, one degree of differencing is done on the data . The results are given below in Fig. 7 & Fig. 8 :

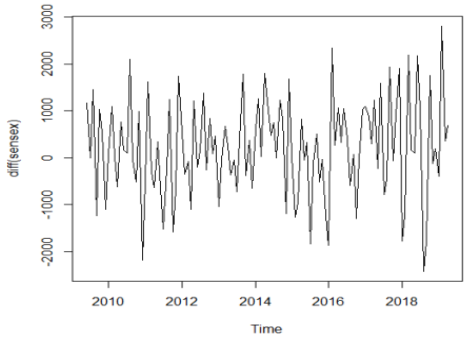


Fig. 7 : Sensex data with d=1 differencing

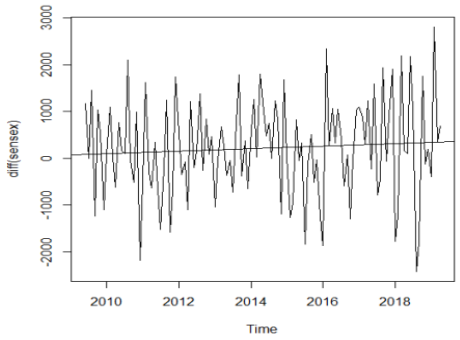


Fig. 8 : Sensex data with d=1 differencing and mean

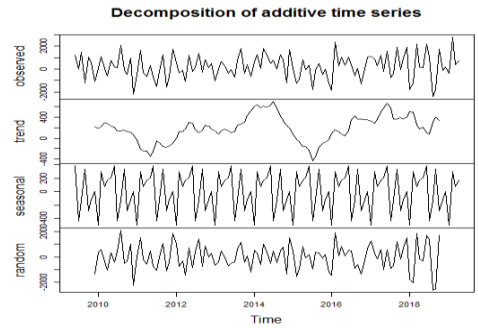


Fig. 9 : Graph representing the decomposed components of the Sensex timeseries after d=1 differencing

It is evident from the Fig. 7 , Fig. 8 and Fig. 9 graphs that the upward trend has dissappeared after the differencing step. Now , the ETS test is performed on the differenced sensex timeseries. The results obtained are given below in Table 3 –

Table 3 : ETS Test Results

Data	Error	Trend	Seasonality
Sensex after differencing with degree 1	Additive	None	None

Source : Results achieved by programming in R

The ADF (Augmented Dickey-Fuller) test is performed in the differenced sensex timeseries to check for stationarity . The confidence level is taken to be 95% . The results obtained are given below in Table 4 –

Table 4 : ADF Test Results

Data	T-Statistic	P-Value	Conclusion
Sensex after differencing with degree 1	-5.1938	0.01	Stationary

Source : Results achieved by programming in R

As is evident from the results of the ETS and the ADF tests , the sensex 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 ARIMA (p,d,q) comes out to be : ARIMA(1,1,2) with drift .

Method 2 : Using the iterative process of choosing the optimal values of p and q (set d=1) for the lowest AIC (Akaike Information Criterion) value gives the following result in Table 5 –

Table 5 : Iterative Test Results for determining value of p and q using AIC

q / p	q=0	q=1	q=2
p=0	1996.222	1996.057	1993.864
p=1	1996.886	1996.926	1993.796
p=2	1992.810	**1992.514**	1994.133

Source : Results achieved by programming in R

The AIC value given by p=2 , q=1 (ARIMA(2,1,1)) is the lowest . Thus using this method the function is ARIMA (2,1,1)

Method 3 : Plot the PACF and the ACF graphs and deduce the vlues and p and q from it.

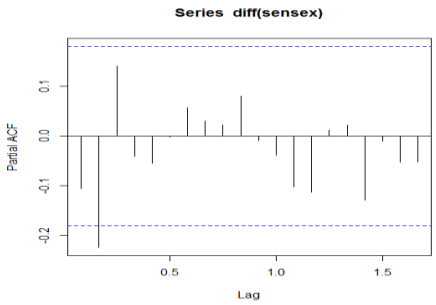


Fig. 10 : PACF graph of the Sensex timeseries after d=1 differencing

It can be seen from this PACF graph in Fig. 10 , that after the spike at lag=1(2nd lag) , the graph cuts off and thus decreases the correlation factor after lag=1 . So p=1 is deduced from the PACF graph.

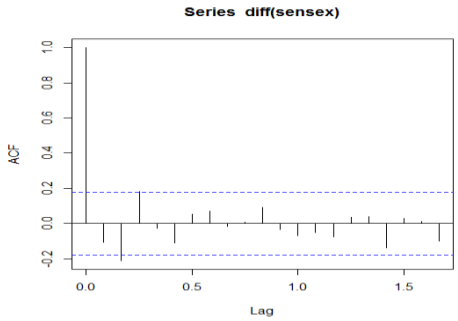


Fig. 11 : ACF graph of the Sensex timeseries after d=1 differencing

It can be seen from this ACF graph in Fig. 11 , that after the spike at lag=2(3rd lag) , the graph cuts off and thus decreases the correlation factor after lag=2 . So q=2 is deduced from the ACF graph.
Thus using this method the function is ARIMA (1,1,2) .

So after implementing all the 3 methods we have 2 Arima function combinations : ARIMA(2,1,1) and ARIMA(1,1,2).
So , 2 predictive models are built - ARIMA(2,1,1) and ARIMA(1,1,2) using which future Sensex monthly closing index value can be predicted . In this paper both these models are used for prediction of sensex monthly closing data for the next 10 years from May,2019 till April,2029 . Two separate data reuslts have been obtained and both have been plotted in Fig. 12 , Fig. 13 , Fig. 14 and Fig. 15 .

```
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
May 2019       40091.32 38777.21 41405.44 38081.56 42101.09
Jun 2019       39939.95 38155.13 41724.76 37210.31 42669.59
Jul 2019       40303.97 38294.04 42313.90 37230.05 43377.90
Aug 2019       40448.60 38183.28 42713.93 36984.08 43913.12
Sep 2019       40686.63 38212.99 43160.26 36903.52 44469.73
Oct 2019       40884.89 38210.95 43558.84 36795.45 44974.34
Nov 2019       41100.09 38243.12 43957.05 36730.73 45469.44
Dec 2019       41308.07 38277.80 44338.34 36673.67 45942.47
Jan 2020       41519.13 38325.47 44712.78 36634.85 46403.40
Feb 2020       41728.87 38379.58 45078.16 36606.58 46851.17
Mar 2020       41939.18 38441.26 45437.10 36589.57 47288.78
Apr 2020       42149.24 38508.72 45789.77 36581.54 47716.94
May 2020       42359.41 38581.68 46137.14 36581.87 48136.96
Jun 2020       42569.54 38659.40 46479.67 36589.50 48549.57
Jul 2020       42779.68 38741.48 46817.88 36603.79 48955.57
Aug 2020       42989.82 38827.49 47152.14 36624.09 49355.54
Sep 2020       43199.95 38917.10 47482.81 36649.90 49750.01
Oct 2020       43410.09 39010.01 47810.17 36680.75 50139.43
Nov 2020       43620.23 39105.96 48134.50 36716.26 50524.20
Dec 2020       43830.37 39204.73 48456.00 36756.07 50904.66
Jan 2021       44040.51 39306.12 48774.89 36799.89 51281.12
Feb 2021       44250.64 39409.96 49091.33 36847.45 51653.84
Mar 2021       44460.78 39516.07 49405.49 36898.50 52023.06
```

Fig. 12 : Portion of the predicted future data starting from May 2019 using ARIMA(1,1,2)

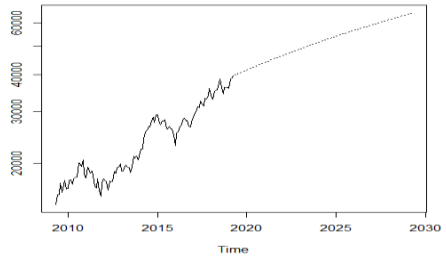


Fig. 13 : Graph plotting the actual data from May 2009 till April 2019 and future predicted data from May 2019 till Apr 2029 using ARIMA(1,1,2) . Solid lines denote actual data and dotted lines denote predicted data .

```
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
May 2019       40190.23 38883.38 41497.09 38191.57 42188.90
Jun 2019       40118.55 38354.80 41882.31 37421.12 42815.98
Jul 2019       40420.58 38445.33 42395.82 37399.70 43441.45
Aug 2019       40660.39 38402.54 42918.25 37207.31 44113.48
Sep 2019       40831.48 38337.85 43325.12 37017.80 44645.17
Oct 2019       41059.62 38364.84 43754.39 36938.32 45180.92
Nov 2019       41274.09 38380.99 44167.19 36849.48 45698.71
Dec 2019       41480.48 38404.41 44556.55 36776.03 46184.92
Jan 2020       41695.35 38447.98 44942.72 36728.93 46661.78
Feb 2020       41907.60 38496.11 45319.08 36690.18 47125.02
Mar 2020       42118.98 38551.34 45686.62 36662.75 47575.21
Apr 2020       42331.60 38614.49 46048.70 36646.77 48016.42
May 2020       42543.74 38682.77 46404.72 36638.89 48448.60
Jun 2020       42755.82 38756.21 46755.42 36638.95 48872.68
Jul 2020       42968.06 38834.48 47101.64 36646.30 49289.82
Aug 2020       43180.23 38916.86 47443.59 36659.97 49700.48
Sep 2020       43392.39 39003.09 47781.70 36679.53 50105.26
Oct 2020       43604.58 39092.85 48116.31 36704.48 50504.68
Nov 2020       43816.76 39185.83 48447.68 36734.36 50899.15
Dec 2020       44028.93 39281.80 48776.06 36768.82 51289.04
Jan 2021       44241.11 39380.56 49101.66 36807.54 51674.68
Feb 2021       44453.29 39481.90 49424.68 36850.20 52056.37
Mar 2021       44665.46 39585.66 49745.27 36896.57 52434.36
```

Fig. 14 : Portion of the predicted future data starting from May 2019 using ARIMA(2,1,1)

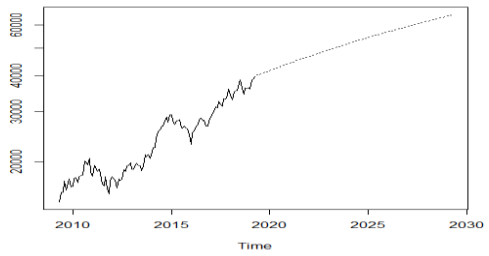


Fig. 15 : Graph plotting the actual data from May 2009 till April 2019 and future predicted data from May 2019 till Apr 2029 using ARIMA(2,1,1) . Solid lines denote actual data and dotted lines denote predicted data .

The 2 predictive models built using the ARIMA technique are now subjected to accuracy testing. The total dataset (May, 2009 – April, 2019) is divided into Training dataset and Testing dataset. The split is done in a 80:20 ratio in favour of Training dataset over Testing dataset. The Training dataset will span from May, 2009 – April, 2017 and the Testing dataset will span from May, 2017 – April, 2019. The two predictive models – ARIMA (1, 1, 2) and ARIMA (2, 1, 1) are trained using the Training dataset. Then they are used to predict the Sensex monthly closing index data from May, 2017 – April, 2019. The results produced are as follows, given in Fig. 16 –

	A	B	C	D	E	F	G	H	I
1		Actual Value	Predicted value with Arima(2,1,1)	Predicted value with Arima(1,1,2)		Actual Value	Predicted value with Arima(2,1,1)	Predicted value with Arima(1,1,2)	
2	May-17	30921.61	32448.66	32449.45	May-18	35423.48	36962.37	36940.68	
3	Jun-17	32514.94	32867.87	32900.66	Jun-18	37606.58	37271.41	37248.33	
4	Jul-17	31730.49	33424.69	33417.86	Jul-18	38645.07	37575.84	37551.54	
5	Aug-17	31283.72	33818.37	33809.97	Aug-18	36227.14	37876.12	37850.57	
6	Sep-17	33213.13	34224.03	34225.68	Sep-18	34442.05	38172.61	38145.88	
7	Oct-17	33149.35	34610.26	34593.73	Oct-18	36194.30	38465.62	38437.73	
8	Nov-17	34056.83	34970.37	34962.43	Nov-18	36068.33	38755.45	38726.43	
9	Dec-17	35965.02	35325.92	35310.28	Dec-18	36256.69	39042.32	39012.20	
10	Jan-18	34184.04	35667.89	35652.79	Jan-19	35867.44	39326.47	39295.26	
11	Feb-18	32968.68	36001.69	35984.15	Feb-19	38672.91	39608.06	39575.79	
12	Mar-18	35160.36	36328.39	36309.55	Mar-19	39031.55	39887.28	39853.95	
13	Apr-18	35322.38	36648.16	36627.75	Apr-19	39714.20	40164.27	40129.91	
14									

Fig. 16: Comparison of the predicted values generated by the 2 ARIMA models in contrast with the actual data value present in the Testing dataset

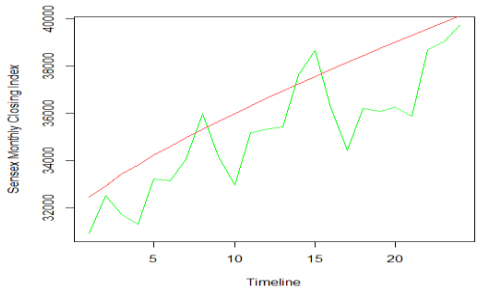


Fig. 17: Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (1, 1, 2)

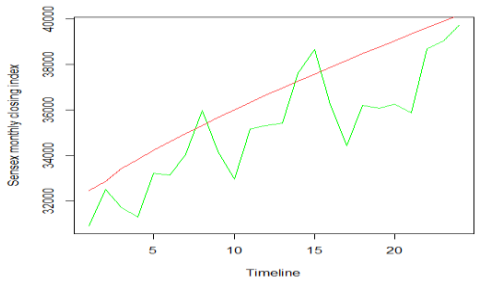


Fig. 18: Line graph representing the Actual Values (Green line) Vs Predicted values (Red line) using ARIMA (2, 1, 1)

In order to perform accuracy testing, the error metrics that have been calculated are MAPE (Mean Absolute Percentage Error) and MPE (Mean Percentage Error). The error metrics observed have been tabulated in Table 6 –

Table 6 : Accuracy testing results

ACCURACY TEST USING ARIMA (1,1,2)		ACCURACY TEST USING ARIMA (2,1,1)	
MAPE (%)	**MPE (%)**	**MAPE (%)**	**MPE (%)**
4.638604	- 4.171705	4.673877	- 4.220957

Source : Results achieved by programming in R

The plotted graphs representing the Actual value Vs Predicted value in Fig. 17 and Fig. 18 show that in predicting stock market closing indices this research paper has been able to achieve a trend and the direction in which the markets are going to move in the future decade.

The accuracy test results show that good and satisfactory values of MAPE and MPE have been achieved. The values indicate >85% prediction success rate.

VIII. CONCLUSION AND SCOPE OF FUTURE IMPROVEMENT

The research paper has tried to study, analyse the Sensex monthly closing data of the last 10 years, and generate a predictive model for forecasting future stock market data. This paper can conclude from the generated predictions and forecasts that there is a upward trend in the sensex data over time and this can be used to make predictions of monthly closing sensex data. The research has a limitation that, while the upward trend has been correctly identified by the study but a more accurate quantification of the predicted data is desirable. Another limitation is that using timeseries analysis; it is very difficult to take into consideration abnormal situations that can affect the stock market like elections , natural calamities etc.

There is a vast scope of future improvement in research in this field. *Artificial neural networks (ANN)* are being used widely now to predict the stock markets for long term capital gains. Facebook has launched an open source library called *Prophet* for analysing and predicting timeseries data. It introduces a new component along with trend, seasonality and error, it is called *holidays*. These two avenues provide a wide scope of future improvement in this field of research related to timeseries forecasting and analysis.

REFERENCES

[1] D. Banerjee, "Forecasting of Indian stock market using time-series ARIMA model", In the proceedings of 2nd IEEE International Conference on Business and Information Management (ICBIM), **pp. 131-135** , **January 2014.**

[2] Ayodele A. Adebiyi, Aderemi O. Adewumi, Charles K. Ayo , "Stock price prediction using the ARIMA model" , In proceedings of the 16th IEEE International Conference on Computer Modelling and Simulation (UKSim) , **pp. 106 -112** , **March 2014** .

[3] A. Edward, J. Manoj, "Forecast model using arima for stock prices of automobile sector" , International Journal of Research in Finance and Marketing , **pp. 1-9** , **April 2016**.

[4] Mohamed Ashik. A, Senthamarai Kannan. K , "Forecasting national stock price using arima model", Global and Stochastic Analysis , **pp. 71-81**, **January 2017**.

[5] M. Angadi, A. Kulkarni , "Time Series Data Analysis for Stock Market Prediction using Data Mining Techniques with R", International Journal of Advanced Research in Computer Science , **Pp. 104-108, August 2015.**

[6] P. Mondal, L.Shit , S.Goswami , "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices", International Journal of Computer Science, Engineering and Applications, **pp. 13-29, April 2014.**

[7] Kamalakannan J , I. Sengupta , S. Chaudhury , "Stock Market Prediction using TimeSeriesAnalysis" , Computing Communications and Data Engineering Series , **Volume No. 01, Issue No. 03 , pp. 1-5 , 2018** .

[8] J.V.N. Lakshmi , Ananthi Sheshasaayee, "A Big Data Analytical Approach for Analyzing Temperature Dataset using Machine Learning Techniques" , International Journal of Scientific Research in Computer Sciences and Engineering , **Volume No. 05 ,Issue No. 03, pp. 92-97 , June 2017**.

[9] Himanshi , Komal Kumar Bhatia, " Prediction Model for Under-Graduating Student's Salary Using Data Mining Techniques" , International Journal of Scientific Research in Network Security and Communication, **Volume No. 06 ,Issue No. 02, pp. 50-53 , April 2018**.

Author's Profile

Debaditya Raychaudhuri has pursued his graduation in Computer Science Honours (BSc.) from St. Xaviers College, Kolkata from 2007 to 2010. He went on to complete his masters (MSc. in Computer Science) from the University of Calcutta in 2012. He worked in Tata Consultancy Services from 2012 till 2013. He qualified UGC Net Examination in June 2013 and December 2013 in Computer Science and Applications. He qualified WBSET Examination in 2013 in Computer science and Applications and achieved the highest marks in the whole state. He qualified GATE in 2014 and achieved an All India Rank of 974. Currently, he is working as Assistant Professor at Chandernagore College, Government of West Bengal since February 2015.His research interest is Machine learning technologies and data analysis.

