Stock Prediction using Autoregressive Integrated Moving Average and Neural Network Auto-Regression

Meenakshi Verma[1]
Shivani Garg[2]

[1]Mtech Student, Department of Computer Science and Engineering, SRCEM at Palwal, Haryana, India.
[2]Assistant Professor, Department of Computer Science and Engineering, SRCEM at Palwal, Haryana, India.

Abstract: The National Stock Exchange (NSE) is the foremost pointer used in the Indian Stock Exchange to calculate the in general progression of the stock market. Precise prediction or forecasting and the progress of the share price or index would profit the shareholder therefore to increase the tactic for effectual stock market trading. Autoregressive Integrated Moving Average (ARIMA) time series model is constructive and intended for best prediction and forecasting on share market and stock price index by using autocorrelation in the time series based data comprising of 1155 days records. However, Neural Network Auto-regression (NAR) is also used to predict the market scenario over the time series dataset. ARIMA with the drift coefficient successfully achieved Training Score of 23.74% using RMSE (Root Mean Squared Error) and Test Score of 11.71% of RMSE (Root Mean Squared Error). ARIMA produces forecasting for 1155 days with upward trend in accordance with the positive drift coefficient. Whereas, Neural Network Auto-regression uses 3 lag auto-regression as input to an artificial neural network system with one hidden layer consisting of 4 units producing NAR forecasting models. Predictions are generated in the testing dataset for 1155 days with RMSE of 29.46%. Evaluation of accuracy in the testing dataset using Mean Absolute Error (MAE) of 867.80; consequently, the scheme proposed that Autoregressive Integrated Moving Average (ARIMA) is the much better technique than Neural Network Auto-regression (NAR) for Stock market prediction.

Keywords: *Machine Learning, Artificial Neural Network, Extreme Learning Machine, Autoregressive Integrated Moving Average, Neural Network Auto-Regression.*

1. INTRODUCTION

The National Stock Exchange (NSE) has experienced tremendous growth as the market develops. The development of the NSE is characterized by high market volatility. This volatility attracts many local and foreign investors because it offers high yield opportunities. The number of companies listed on the NSE increased to 1795 in 2020 from 1438 in 2018. The total trading volume reached 43,406 billion shares with a total market capitalization of US $. 2.9 trillion in 2029 (NSE, 2020). The Stock Price Index or Indices is the main indicator used on the NSE to measure the overall stock market performance. NSE is a weighted market capitalization index and represents at least 70% of the total market capitalization and number of shares traded.

The basic theory of the stock market especially related to stock market forecasting is the market-efficient hypothesis put forward by Eugene Fama in 1965. Fama (1965) [3] states that the stock market is very efficient at processing information and adapting so quickly to new information that prices are at all times already reflect all available information. Price movements in the stock market are strongly influenced by new information, and new information is basically random. Therefore each observation in the time series of price movements should be random and not correlated with each other or independently. The markets where each movement is independent are markets with a random walk process. According to the random walk theory, the best forecast for stock prices is a function of the current price plus a random error. As a result, analytical methods that use historical stock trading data are not useful for forecasting.

There are various approaches used to forecast the stock market. Technical analysis is a stock market analysis method for forecasting price movements using historical stock trading data, especially price and volume data [2]. This method emphasizes the use of charts and stock market indicators to detect stock price movements. Technical analysis rejects the efficient market hypothesis and suggests that price movements form trends and repetitive patterns that can be predicted. The rationale for technical analysis is that all factors that can affect stock prices, whether economic, political, or investor psychology, will be reflected in price and volume.

Autoregressive Integrated Moving Average (ARIMA) is a linear time series model that is useful both to understand the time series process and to produce forecasting [6]. This model is a univariate time series model that projects or extrapolates the historical values of the predicted variables by identifying past patterns contained in the data.

ARIMA consists of two components, namely the autoregressive model and the moving average model. Auto-regression models the autocorrelation of time series variables that depend linearly on the values of the previous variables. The moving average model models the autocorrelation of previous errors contained in time series data [4]. ARIMA is a generalization of the ARMA model that can be applied to non-stationary time series data through time series differentiation.

Several studies have been carried out to forecast stock prices using ARIMA [5] estimated the forecasting model on the composite stock price index for five Asian countries including India [4] conducted a study to find out the behaviour of price movements on the stock market by conducting several estimates of the stock market forecasting models including ARIMA. The results of this study indicate that ARIMA is a useful model for forecasting in the stock market.

Neural Network Auto-Regression is a time series forecasting model that uses lag variables as input into an artificial neural network system [6]. Artificial neural network (ANN) is an excellent model for solving nonlinear problems and has been applied in a variety of applications from all fields including for stock market predictions [5]. ANN is a nonparametric model that does not depend on assumptions such as stationary, normality, and heteroscedasticity, so it can be used to identify complex nonlinear relationships between predictor variables and response variables with high accuracy.

To find out whether stock price movements on the stock market follow the random walk process and in accordance with the weak form-efficient market hypothesis or there are other processes that can be modelled for forecasting, this study aims to produce and compare two forecasting models namely Autoregressive Integrated Moving Average ARIMA, and Neural Network Auto Regression. The number of auto-regression lags produced is then used as input into the feed-forward / back-propagation neural network system with one hidden layer to produce NAR forecasting. The accuracy of forecasting ARIMA and NAR is then measured and compared using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) [7-9].

2. LITERATURE REVIEW

Prediction is the process of forecasting the occurrence of a scientific event next time. In the planning process, predictions become stages first of the process. The prediction is done using data from the past of one or more variables to estimate the value of the future come. Prediction is needed by almost all agencies in this case electricity provider to make decisions related to the amount of power electricity that must be generated in the future. Effective prediction much needed to achieve the strategic and operational goals of all institution or industry. At electrical energy providers, predictions are made for estimate the demand (load) or load (load) of electrical energy. Forecasts can control the production control system (generation) and distribution based on needs. In the public sector, forecasts are a part of inseparable from the design of policies and programs, both in the field economy, education and public health. [10]

Prediction techniques can be divided into several types depending on the angle his views, including:

1. Based on the nature of the Predictor:-
    a. Subjective predictions, i.e. predictions that make the subject or predictor as a major determinant of whether or not the results of these predictions.
    b. Objective predictions, i.e. predictions based on relevant data the past by using certain techniques or methods.
2. Based on Prediction Period:-
    a. Short-term predictions, i.e. predictions made to predict a state that will appear every hour until one week. Short-term predictions are usually used to get comparisons between forecasts and real-time conditions.
    b. Medium term forecasting, i.e. prediction carried out for a period of one week to one year.
    c. Long-term predictions (long-term forecasting), i.e. predictions that are done to predict a situation in the future within the next few years.
3. Based on the Nature of Prediction:-
    a. Qualitative predictions, namely predictions based on qualitative data in the past.
    b. Quantitative predictions, i.e. predictions based on quantitative data in the past. The results of quantitative predictions will be very depends on the method used in the prediction.
    c.Neural Network Artificial neural network or often called simply with a neural network (Neural Network) has been widely has been developed as a study of electrical load prediction techniques since 1990. Outputs of an artificial neural network are several linear mathematical functions or non-linear value of the input. The input value can be output from other network elements such as the actual network input.

Artificial Neural Networks: [11] Explains how artificial neural networks can be used in the process of data mining, with characteristics of artificial neural networks. Especially information about the rules contained can be predicted and explained from the data. With the final result is that back-propagation neural networks need to be given some limiting functions so that the process can be directed for the purposes of the formation of logical relations. The result of applying the limiting function shows that the first process of learning is not interrupted and the extraction of logical values can be done. Extraction results show that by forcing learning weights towards absolute values (0.1, and -1) it gives good results.[12] Conducted a study of forecasting annual electricity consumption using artificial neural networks in the industrial sector. Chemical material, base metals and non-industrial mineral metals are defined as industries that consume high energy. Regression in conventional models does not estimate energy consumption correctly and precisely. Although artificial neural networks are commonly used to predict consumption in the short term; this research shows that a more appropriate approach to predict annual consumption in the industry. Artificial neural networks using an approach based on supervised multi-layer perceptrons are used to show estimates of annual consumption with small errors.

Artificial neural network (ANN) is a paradigm of processing information that is inspired by the biological nerve cell system. ANN is formed as a generalization of mathematical models of biological neural networks, assuming that:

1. Information processing occurs in many simple elements (neurons).
2. Signals are sent between neurons through links.
3. The connection between neurons has a weight that will strengthen or weaken the signal.
4. To determine output, each neuron uses an activation function (usually not a linear function) that is imposed on the sum of the inputs received. The amount of output is then compared to a threshold.

Neurons biology is a system that is "fault tolerant" in 2 ways. First, humans can recognize input signals that are somewhat different from those we have ever received before. For example, humans can often recognize someone whose face has been seen from a photo or can recognize someone whose face is somewhat different because it's been a long time. Second, still able to work well. If a neuron is damaged, other neurons can be trained to replace the function of the damaged neuron [13].

The thing to be achieved by training ANN is to strike a balance between memorization and generalization capabilities. What is meant by memorization ability is the ability of ANN to take back perfectly a pattern that has been learned. The ability to generalize is the ability of ANN to produce an acceptable response to patterns that have been studied. This is very useful if at one time into the ANN it is inputted with new information that has never been studied, then the ANN will still be able to give a good response, giving an approaching output [14].

Artificial neural networks resemble the human brain in 2 ways, namely:

1. Knowledge is gained by the network through the learning process.
2. The strength of the relationship between nerve cells (neurons) known as synaptic weights is used to store knowledge. Artificial neural networks are determined by 3 things.
   a. Patterns of relationships between neurons (called network architecture).
   b. Method fordetermineweightconnector (called method training / learning).
   c. Activation function, which is a function used to determine an output neurons.

Back-propagation Network Training: Back-propagation network training rules consist of 2 stages, feed-forward and backward propagation. In the network given a set of training examples called training sets. This training set is illustrated by a feature vector called an input vector that is associated with an output that is the target of the training. In other words the training set consists of an input vector and also a target output vector. The output of the network is an actual output vector. Then do a comparison between the actual output produced with the target output by reducing the two outputs. The result of the reduction is an error. Error is used as the basis for making changes to each weight by reproducing it again. Every change in weight that occurs can reduce errors. The weight change cycle epoch) is carried out on each training set so that the conditions stop is reached, i.e. if it reaches the desired number of epochs or until a specified threshold value is exceeded. The back-propagation network training algorithm consists of 3 stages that is [15,16]:

1. Forward feed stage (feed-forward).
2. Feedback stage (back-propagation).
3. The updating stage of weights and biases.

In detail the back-propagation network training algorithm can be described as follows:

Step 0: Initialize weights, training rate constants (α), tolerance errors or weight values (when using weight values as a stop condition) or set the maximum epoch (if using the number of epochs as a stop condition).

Step 1: As long as the stop condition has not been reached, then do step 2 to step 9.

Step 2: For each pair of training patterns, do steps 3 through step 8.

Step 3: {Stage I: Feed-Forward).

Each input unit receives a signal and passes it to the hidden unit above it.
Step 4: Each unit in the hidden layer (from unit 1 to unit p) is multiplied by its weight and added up and added to its bias.
Step 5: Each unit of output (yk, k = 1,2,3, ... m) is multiplied by weight and added up and added to the bias.
Step 6: {Stage II: Backward propagation}.
Each output unit (yk, k = 1,2,3, ... m) receives the target pattern tk according to the input pattern during the training and then the output layer (δk) error information is calculated. δk is sent to the layer below and used to calculate the magnitude of weight and bias correction (ΔWjk and ΔWok) between the hidden layer and the output layer.
Step 7: Each hidden layer unit (from unit 1 to p; i = 1 ... n; k = 1 ... m) calculates hidden layer error information (δj). δj is then used to calculate the magnitude of weight and bias correction (jiVji and ΔVjo) between the input layer and the hidden layer.
Step 8: {Stage III: Updating weights and biases}.
Each unit of output (yk, k = 1,2,3, ..., m) is updated its bias and weight (j = 0,1,2, ... p) so as to produce new weights and biases. Likewise for each hidden unit starting from the 1st unit to the pth unit, weights and biases are updated.
Step 9: Test the stop condition (end of iteration).
Stock Price Index:Stock price index is an indicator or reflection of the price movement of a group of shares. The stock price index is one of the guidelines for investors to invest in the capital market. Generally there are two methods of calculating the stock price index, namely [17-20]:

1. Weighted Price Index

The weighted price index calculates the index number by dividing the average stock price contained in the index composition at time t, by the average price at the base time (base value). This method uses prices as a weighted average for index calculation as follows:

$$\text{Weighted Price Index} = \frac{1}{\text{n}}\sum_{1=1}^{n}\frac{P_{i,t}}{P_{i,basic}} * 100 \quad (eq.1)$$

Where:

$P_{i,t}$ = stock price i at time t
N = number of shares incomposition

2. Weighted Market Capitalization Index

Weighted market capitalization index calculates a weighted average based on market capitalization. The index number is obtained by dividing the market capitalization value of shares contained in the composition of the index at time t, by the value of market capitalization at the base time, where market capitalization is the market price times the number of shares. Weighted market capitalization index calculation:

$$\text{Market} - \text{Capitalization Index Weighted} = \frac{\sum_{i=1}^{n} P_{i,t} Q_{i,t}}{\sum_{i=1}^{n} P_{i,basic}\, Q_{i,t}} \quad (eq.2)$$

Where:
$P_{i,t}$ = stock price i at time t
$Q_{i,t}$ = number of shares i at time t
N = number of shares in composition

Autoregressive Moving Average (ARMA): When data has autocorrelation, most forecasting methods based on the assumption of independent observations are invalid. Stationary time series with autocorrelation can be modelled using Autoregressive Moving Average. ARMA consists of two components, namely autoregressive and moving average [21]. Autoregressive (AR) is a time series model that regresses the value of past observations, namely the lag variable. Auto-regression with lag p as MA (q), given with [22,23]:

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + \varepsilon_t \quad (eq.3)$$

Where,
c= constant
Φ = model parameters
y_{t-p} = lag variable p
ε_t= period error t
Moving Average (MA) is a time series model that performs a regression of lagged errors to produce forecasting. Moving Average lag q, MA (q), given with:-

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_2 \varepsilon_{t-2} + \theta_q \varepsilon_{t-q} \quad (eq.4)$$

Where,

θ = model parameters
ε_{t-q} = error in t lag period q
Autoregressive (AR) and Moving Average (MA) can be effective combined to form a time series model known as the ARMA model. ARMA (p, q) is given with:

$$y_t = c + \Phi_1 y'_{t-1} + \Phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (eq.5)$$

***Autoregressive Integrated Moving Average (ARIMA)*:** ARIMA is a time series model that utilizes differentiation in the ARMA model so that it can be used for modeling non-stationary time series. ARIMA uses time series differentiation with level d to produce stationary time series. The ARIMA model (p, d, q) is given as [48-51]:

$$y'_t = c + \Phi_1 y'_{t-1} + \cdots + \Phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (eq.6)$$

The three ARIMA components appear clearer when written using the backshift operator, $By_t = y_{t-1}$, as follows:

$$(1 - \Phi B - \cdots - \Phi_p B^p) y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t \qquad (eq.7)$$

P = order autoregressive
D = level differentiation
Q = order moving average
ARIMA can be used for modeling various time series processes that are both stationary and non-stationary because the ARIMA model combines AR, MA and differentiation components. Table 2.1 gives the equivalence of time series processes with the ARIMA model (p, d, q).

Process	**ARIMA (p, d, q)**
White Noise	ARIMA (0,0,0)
Auto Regression	ARIMA (p, 0,0)
Moving Average	ARIMA (0,0, q)

Table 2.1 Time series processes in the form of ARIMA.

Neural Network Auto-Regression (NAR):Neural Network Auto regression (NAR) is a simple hybrid model that combines auto regression and artificial neural networks. For univariate time series forecasting, the value of the lag variable can be used as input to the Artificial Neural Network (ANN) system to produce nonlinear auto regression, almost the same as in the auto regression model where the lag variable is used in linear regression [24,25]. ARIMA approach to prediction the time series assumes that the data under study is generated from a linear process. ANN is a data-driven and self-adaptive method where the resulting model is determined by the characteristics contained in the data itself. The basic idea of this multi-model approach is to utilize the unique abilities of each component of the model to better capture different patterns in the data. Regression models that use neural fuzzy. Suppose an input pair is given - output (x_k, d_k), k = 1,2, ..., p with x_k = (x_{k1}, x_{k2}, ..., x_{kn}). A regression model in the k-th pattern is represented as [53 ,55]:-

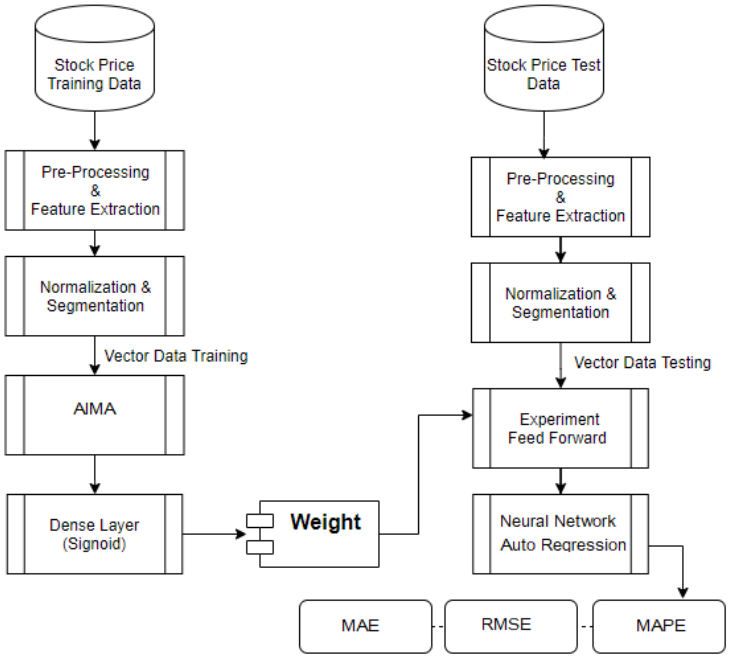
$$Y(x_k) = A_0 + A_1 x_{k1} + \ldots + A_n x_{kn} \quad (eq.8)$$

where A_n time series number. Therefore, the estimated value of output Y (xk) is also a regressed number. Regression analysis can be simplified into interval regression analysis where the interval regression model will be formed regression model is the development of classical regression in which some elements such as input or output or both are random numbers. The basic concept of regression analysis is based on back-propagation networks uses 2 back-propagation networks. One network is used for the upper limit of the interval, while one other network is used for the lower limit of the data interval. Both networks are trained separately For example, g + (xk) and g- (xk) are the outputs of the two back propagation networks (BPN + and BPN-) associated with xk vector input, where each network has n neurons in the input layer and 1 neuron.

$$g(x) \le d_k \le g^+(x), k = 1,2,\ldots.,p) \quad (eq.9)$$

3. PROPOSED WORK

***Research Flow Chart*:**The research flow diagram is given in Figure 1 below.



***Research Data and Variables*:** The research variable is univariate time series y1 , y_2, ...,y_n That is a sample of NSE daily closing price data for the January 2015 period to December 2019. Secondary data collection is done by downloading data from http://finance.yahoo.com. NSE is a market-capitalization index.

***Data Processing*:** Data processing is performed to check whether there are outliers and missing data. Data imputation is then performed to produce a complete initial dataset. The data is then divided using the hold out method as in Figure2 below.

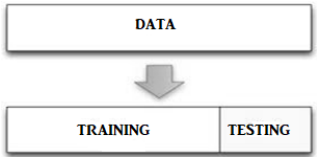


Figure 2 Hold Out Method.

(1) 80% for the training data set used in the estimation model.

(2) 20% for the test data set used in evaluating the accuracy of the model.

***Modelling*:** The flow diagram in Figure 1 adopts the stages carried out in modelling. The modelling stages are: (1) Identification Stage, (2) Estimation Stage, (3) Diagnostic Stage. After an appropriate ARIMA model is obtained, forecasting for the next few periods can be generated. The last step is evaluating forecast accuracy by measuring the error between the predicted value and the actual value.

***Identification Stage*:** The first stage in modelling is determining whether the time series is stationary and if there are significant trends that need to be modelled. The basis for any time series analysis is the assumption that time series are stationary in mean and variance. Stationary was detected visually using time series plots, autocorrelation function plots, and by using the augmented stationary test. Non-stationary time series will show significant co-relation that decreases very slowly can be then carried out to produce a stationary time series in mean and variance.

***Estimation Stage*:**Maximum likelihood estimation (MLE) is a parameter estimation method that maximizes the probability of the model generated accordingly with data. Estimated parameters $c, \Phi_1, \dots, \Phi_p, \theta_1, \dots, \theta_q$ for the ARIMA model are obtained by minimizing:

$$\frac{1}{2T}\sum_{t=1}^{T}\varepsilon_t^2 = \frac{1}{2T}\sum_{t=1}^{T}(y_t - c - \Phi_1 y_{1,t} - \cdots - \Phi_k y_{k,t})^2 \quad (eq.10)$$

Information Criterion (IC) is a goodness of fit criterion used to measure how well the estimated model fits the data. The p and q orders that produce the best ARIMA models are those that have the smallest IC. IC for the ARIMA model is given as:

$$IC = -2\log(L) + 2\,(p + q + k + 1)\ \ (eq.11)$$

Where,

L = data likelihood

K=1 if c ≠0 then k=0 then c=0

Estimation is done using the help of a computer program that implements the optimal ARIMA modeling algorithm automatically.

The algorithm in estimating the ARIMA model is as follows:

1. Determine the initial four models as follows:
    1. ARIMA (0, d, 0)
    2. ARIMA (1, d, 0)
    3. ARIMA (0, d, 1)
    4. ARIMA (2, d, 2)
2. Perform parameter estimates $c, \Phi_1, \ldots, \Phi_p, \theta_1, \ldots, \theta_q$ with Maximum Likelihood Estimation.
3. Calculate Information Criterion.
    1. The best model is the model with the smallest IC.
    2. If d = 0, then set c ≠ 0. If d ≥ 1, then set c = 0.
    4. Try variations of p and q of ∓ 1 on the best model.
    5. Repeat steps 2-4 until there is no smaller IC.

***Diagnostic Stage*:** The diagnostic stage is carried out to check whether the resulting ARIMA model meets the following requirements:

1. Has significant parameters.
2. Residuals have zero average and do not correlate.
3. Residuals are normally distributed.

Test the Significance of Parameters:Parameter significance tests are carried out to check whether the parameters produced at the estimation stage are significant based on: Hypothesis:

Hypothesis:

$H_0: parameter = 0$

$H_a: parameter \neq 0$

Test Statistics:

$$t = \frac{\hat{\Phi}_p}{SE(\hat{\Phi}_p)}\ where\ t = \frac{\hat{\theta}_p}{SE(\hat{\theta}_p)}\ (eq.12)$$

Rejection area:

RejectedH_0if |t| >$t_{\frac{\alpha}{2},n-p}$ or p-value <α of 5% .

N = number of observations

P = amount parameter

Residual Assumption Test: The test is carried out to check whether the residual model meets the noise assumption, so there is no bias and no more information can be modelled. A time series that has no autocorrelation is called "white noise". White noise is a stationary time series process with E (μ) = 0 and the expected autocorrelation coefficient approaches zero and is inside the critical value limit at $\mp 2/\sqrt{T}$ test is based on hypothesis:-

H_0: r_i = 0, i = 1, 2, ..., K (residual is white noise)

H_a: There is at least one $r_i \neq 0$, ι = 1,2, ..., K

Test Statistics:

$$Q = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k\ \ (eq.13)$$

Rejection area: Reject H_0 if $Q^* > x^2_{a,h-K}$ or p-value <α .

Arima Forecasting: After estimating the ARIMA model, forecasting for one or several periods in the future can then be made. The result of forecasting is the estimated value of the random variable to be predicted (point forecast). ARIMA forecasting can be accompanied by prediction intervals that provide a range of variable values with a certain probability (interval forecast). Forecasting intervals assume that forecast error has no correlation and is normally distributed. Calculation of forecasting interval is given by:

$$\hat{y}_t = \mp t_{a/2}\hat{\sigma}\quad (eq.14)$$

Where,

$\hat{y}_t$= estimate of y at time t

$t_{a/2}$= value of t with confidence interval
$\hat{\sigma}$=standard residual deviation

Neural Network Auto-Regression (NAR) Modeling: Neural Network Auto-regression use the lag variable as input into the neural network system; the steps in NAR modeling are as follows:

1) ***Lag Variable Data Processing:*** For NAR modeling, the processing method used is the same as in ARIMA modeling, but further processing is done to produce a dataset that includes lag variables and in accordance with the format used by computer vision and natural language processing packages in the python program. Normalization of data is then performed to produce data with zero average and standard deviation one. It is intended that the back-propagation algorithm can more quickly converge to produce weights with the smallest error.

2) ***Determine Network Topology:*** The amount of lag from the AR order resulting from the ARIMA estimation is used as input into the Multilayer Feed-forward system with one hidden layer and one output layer. This ANN system models the relationship between input variables are lag variables $y_{t-1}, y_{t-2}, y_{t-3}$ and the output variable y_t There is no Standard rules for determining the number of units in the hidden layer and using the same number of units as the input layer will produce a fairly adequate model. The specified number of units in the hidden layer is the number of units in the input layer plus one unit.

3) ***Estimation and Diagnostics:*** The back-propagation algorithm is run to produce the estimated parameters of the model with the smallest error. Residual diagnostics are then performed to check whether the resulting error has an average of zero. After satisfactory diagnostic test results, the model can be used for predictions on the testing dataset.

***Evaluation Of Forecasting Accuracy*:** Error Forecasting is the difference between the predicted values with true value and given with, measured using $e_i = y_i - \hat{y}_i$ Accuracy of forecasting models using:

1. Mean Absolute Error (MAE):$MAE = Mean(|e_i|)\ (eq.15)$
2. Root Mean Squared Error (RMSE):$RMSE = \sqrt{mean(e_i^2)}\ \ (eq.16)$
3. Mean Absolute Percentage Error (MAPE):

The error percentage is given with $p_i = 100\ e_i / y_i$ MAPE has the advantage that it does not depend on the scale of the data so that it can be used to measure several models that use data at different scales. MAPE is calculated by:$MAPE = mean(|p_i|)\ \ (eq.17)$

4. RESULTS AND SIMULATION

Implementation: This section explains approval data and then forecasting data with the Method ARIMA and NAR method are specialized in the field forecasting. After analyzing it will compare the RMSE and MSE value of the results forecasting using both methods so that it can be known which method is best used in forecasting the Stock Price Index. However, the movement of the Stock Price Index during the observation period showed an upward trend with no seasonal patterns. There are several sub-periods with a strong upward trend and several sub-periods where the price fluctuates.

Data Collection: In determining the forecasting results with the ARIMA Method and the Method NAR used Stock Price Index data for 1355 rows that is, starting from January 2016 to December 2019. Data obtained from the official website of the National Stock Exchange using yahoo finance.

rawData.csv

Sno,Close	11,860.080017
0,950.2800289999999	12,856.51001
1,954.7199710000001	13,853.98999
2,948.4500119999999	14,855.1300050000001
3,937.090027	15,840.179993
4,932.820007	16,841.4600220000001
5,924.5200199999999	17,839.8800050000001
6,891.4400019999999	18,841.7000119999999
7,889.1400150000001	19,842.099976
8,888.840027	20,845.099976
9,878.929993	21,848.909973
10,858.9500119999999	22,852.570007
11,860.080017	23,856.75

Figure3: Raw Data Comprising of 1355 Rows Pertaining the Closing Price of each Day.

***Model Estimation–ARIMA*:** Estimation using the python program produces an ARIMA with a positive drift coefficient. This means that autocorrelation with lag 30 and errors with lag density 1 can be used to forecast share index or price movement. A positive drift coefficient indicates an upward trend. Estimated outputs and coefficients are given in Figure 4 using the code block in Figure 5.

```
# create and fit the ARIMA network
model = Sequential()
model.add(LSTM(30, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='arima')
model.fit(trainX, trainY, epochs=10, batch_size=1, verbose=2)

# make predictions
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
# invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
# calculate root mean squared error
trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print('Test Score: %.2f RMSE' % (testScore))

print("--- %s seconds ---" % (time.time() - start_time))
```

Figure 4. ARIMA Network Model Call

Training Result of ARIMA Network Model: In the table 1 goes up indicates a trend in the data using training model, so that there is a clear trend cause the data is not stationary. For this reason, before it is done more with ARIMA, it needs to be done differentiation process. The next step that must be done is by determine the value of d (differencing)or differentiator his is in accordance with the principle of parsimony which always tries to choose a model simple. Thus the number d in the ARIMA model (p, d, q) becomes 1, so it can be identified that the data can be used in the ARIMA model (p, d, q). This stage will be carried out during the parameter estimation and diagnostic process check which result the below matrix as training result.

Sno.	*Train-X*	*Train-Y*	*Sno.*	*Train-X*	*Train-Y*
0	*950.28*	*858.95*	*5*	*924.52*	*840.18*
1	*954.72*	*860.08*	*6*	*891.44*	*841.46*
2	*948.45*	*856.51*	*7*	*889.14*	*839.88*
3	*937.09*	*853.99*	*8*	*888.84*	*841.7*
4	*932.82*	*855.13*	*9*	*878.93*	*842.1*

Table 1: Training Results achieved using ARIMA

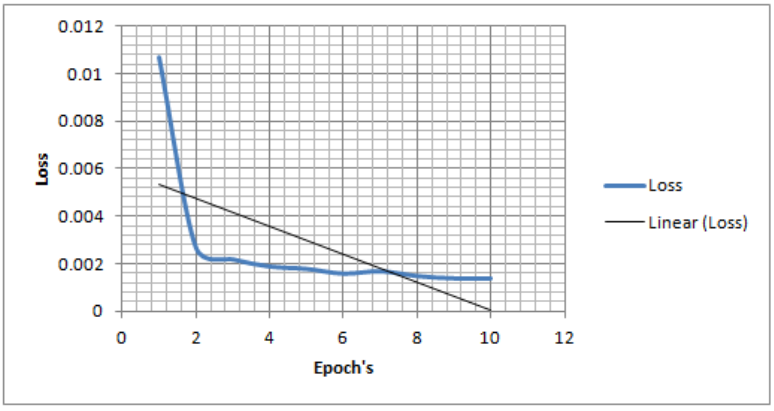


Figure 4: Graph Representation of Training Results Achieved in Table 1

```
Epoch 1/10
 - 7s - loss: 0.0107
Epoch 2/10
 - 5s - loss: 0.0027
Epoch 3/10
 - 5s - loss: 0.0022
Epoch 4/10
 - 5s - loss: 0.0019
Epoch 5/10
 - 5s - loss: 0.0018
Epoch 6/10
 - 5s - loss: 0.0016
Epoch 7/10
 - 5s - loss: 0.0017
Epoch 8/10
 - 5s - loss: 0.0015
Epoch 9/10
 - 6s - loss: 0.0014
Epoch 10/10
 - 7s - loss: 0.0014
Train Score: 23.74 RMSE
Test Score: 11.71 RMSE
650.02002045 659.01000938 668.28002362 665.40999053 645.9000237
  642.40003064 639.70002984 640.24999489 633.14001577 629.7000189
  625.82000689]
```

Figure 5: RMSE - Train Score of 23.74% and RMSE Test Score of 11.71% using ARIMA with Price Prediction of Index Rs. 625.82 of next opening day.

Neural Network Auto-Regression (NAR) Modeling: The function of binary sigmoid (log sigmoid), is used as activation functions in the first hidden layer, second hidden layer and output for NAR. However, Neural Network Auto-regression uses 3 lag auto-regressions as input to an artificial neural network system with one hidden layer consisting of 4 units for producing NAR forecasting models.

```
# create and fit Multilayer Perceptron model using NAR
model = Sequential()
model.add(Dense(12, input_dim=look_back, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=10, batch_size=2, verbose=2)
# Estimate model performance
trainScore = model.evaluate(trainX, trainY, verbose=0)
print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
#changes here
#testScore = model.evaluate(testX, testY, verbose=0)
#print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))
print(numpy.array(list_needed))
# generate predictions for training
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
```

Predicting Stock Price Using NAR

Batch No.1	NSE	
Epoch's	10 Nos	
Epoch No.	Time Elapsed	Loss
1/10	2 Seconds	53245.059
2/10	2 Seconds	1631.8918
3/10	2 Seconds	1391.3422
4/10	2 Seconds	1283.3467
5/10	2 Seconds	1220.7342
6/10	2 Seconds	1121.6011
7/10	2 Seconds	1054.0103
8/10	2 Seconds	953.8202

9/10	2 Seconds	971.9443
10/10	2 Seconds	934.0731
Train Score: 867.80 MSE (29.46 RMSE)		

Table 2 Results using NAR

```
Epoch 1/10
 - 2s - loss: 53244.5365
Epoch 2/10
 - 2s - loss: 1631.8742
Epoch 3/10
 - 2s - loss: 1391.3408
Epoch 4/10
 - 2s - loss: 1283.3409
Epoch 5/10
 - 2s - loss: 1220.7321
Epoch 6/10
 - 2s - loss: 1121.5961
Epoch 7/10
 - 2s - loss: 1054.0083
Epoch 8/10
 - 2s - loss: 953.8191
Epoch 9/10
 - 2s - loss: 971.9417
Epoch 10/10
 - 2s - loss: 934.0725
Train Score: 867.80 MSE (29.46 RMSE)
[659.01 668.28 665.41 645.9  642.4  639.7  640.25 633.14 629.7 ]
```

Figure 6: RMSE – Test Score of 29.46% and MSE Test Score of 867.80% using NAR with Price Prediction of Index Rs. 629.70 of next opening day.

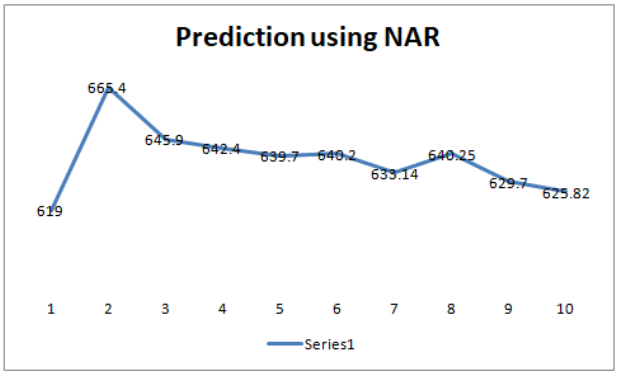


Figure 7: Chart Representation of Result Achieved using NAR

5. CONCLUSION AND FUTURE SCOPE

Conclusion: From ARIMA modelling results for forecasting the Stock Price Index, the conclusions that can be drawn are:

1. Stock Price Index during the observation period shows an upward trend and significant autocorrelation. The movement of stock prices on the National Stock Exchange (NSE) does not follow the random walk process and is not in accordance with the market-efficient hypothesis.
2. Estimation of the model in the training data yields an ARIMA with the drift coefficient as follows: with Training Score of 23.74 using RMSE (Root Mean Squared Error) and Test Score of 11.71 of RMSE (Root Mean Squared Error).
3. ARIMA produces forecasting for 1155 days. Point forecast results in forecasting in the form of an upward trend in accordance with the positive drift coefficient. The forecast interval with an interval of 80% and 95% results in a widening forecasting interval.
4. Neural Network Auto-regression uses 3 lag auto-regression as input to an artificial neural network system with one hidden layer consisting of 4 units producing NAR forecasting models. Predictions are generated in the testing dataset for 1155 days with RMSE of 29.46%. Evaluation of accuracy in the testing dataset using Mean Absolute Error (MAE) of 867.80 shows that the NAR model produces the highest forecasting error with the range of predicted values in figure7; The results of this accuracy evaluation indicate that NAR appears to be over fitting and lacks good generalizations.

5. ARIMA forecasting in the form of trends would be more suitable for forecasting in the stock market which is very volatile and full of uncertainty. This is in accordance with the saying that "The Trend is your Friend, Until It Bends" as the RMSE and MSE is in range and prediction is appropriate as mentioned in figure 6.

Suggestions: Some suggestions that can be given to improve forecasting and development models ARIMA & NAR.

1. Increase the amount of data and use the Walk-Forward Optimization method.
2.Improving Artificial Neural Networks (ANN) to produce forecasting models with better generalizations.
3. Using Vector Auto regression (VAR) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH).
4. Develop automated trading strategies based on forecasting models.

REFERENCES

1. https://www.nseindia.com
2. Mansor, Rosnalini&Zaini, Bahtiar&Yusof, Norhayati. (2019). Prediction stock price movement using subsethood and weighted subsethood fuzzy time series models. AIP Conference Proceedings. 2138. 050018. 10.1063/1.5121123.
3. Clement, Douglas. (2007). Interview with Eugene Fama. The Region. 15-23.
4. Akuffo, Buckman&MintahAmpaw, Enock. (2020). An Autoregressive Integrated Moving Average (ARIMA) Model For Ghana's Inflation (1985 -2011).
5. Borkin, Dmitrii&Németh, Martin &Nemethova, Andrea. (2019). Using Autoregressive Integrated Moving Average (ARIMA) for Prediction of Time Series Data. 10.1007/978-3-030-30329-7_42.
6. Patulin, Elvis. (2019). Crime Prediction using Autoregressive Integrated Moving Average (ARIMA) Algorithm. International Journal of Advanced Trends in Computer Science and Engineering. 8. 720-724. 10.30534/ijatcse/2019/59832019.
7. Wu, W., An, S., Guan, P. et al. Time series analysis of human brucellosis in mainland China by using Elman and Jordan recurrent neural networks. BMC Infect Dis 19, 414 (2019). https://doi.org/10.1186/s12879-019-4028-x
8. https://otexts.com/fpp2/accuracy.html
9. RatnadipAdhikari, R. K. Agrawal , Thesis on An Introductory Study on Time Series Modeling and Forecasting, JNU, New Delhi , https://arxiv.org/ftp/arxiv/papers/1302/1302.6613.pdf
10. Febi Satya Purnomo, "The Use of the ARIMA (Autoregressive) Method Integrated Moving Average) For Electricity Load Estimates Short Term (Short Term Forecasting) ", Journal of UNS, Semarang, 2015
11. Cappy, Alain. (2020). Artificial Neural Networks. 129-210. 10.1002/9781119721802.ch4.
12. Bell, Jason. (2020). Artificial Neural Networks. 10.1002/9781119642183.ch9.
13. Conde-Gutiérrez, RA & Cruz Jacobo, Ulises & Perez, J A. (2020). Use of Artificial Neural Networks in. 10.1201/9780429436963-21.
14. Singh, Himanshu& Lone, Yunis. (2020). Artificial Neural Networks. 10.1007/978-1-4842-5361-8_5.
15. Li, Hongyan& Tian, Qi & Wang, Xiaojun& Wu, Ya'nan. (2014). Multivariate Coupling Sensitivity Analysis Method Based on a Back-Propagation Network and Its Application. Journal of Hydrologic Engineering. 20. 06014013. 10.1061/(ASCE)HE.1943-5584.0001131.
16. Li, Hongyan& Tian, Qi & Wang, Xiaojun& Wu, Ya'nan. (2014). Multivariate Coupling Sensitivity Analysis Method Based on a Back-Propagation Network and Its Application. Journal of Hydrologic Engineering. 20. 06014013. 10.1061/(ASCE)HE.1943-5584.0001131.
17. Gao, Zhiyuan& Qi, Likai. (2010). Predicting Stock Price Index.
18. Aryasta, I &Artini, Luh. (2019). The Effects of Indonesian Macroeconomic Indicators and Global Stock Price Index on the Composite Stock Prices Index in Indonesia. International Journal of Scientific and Research Publications (IJSRP). 9. p9069. 10.29322/IJSRP.9.06.2019.p9069.
19. McIntyre, Francis. (2020). The Problem of the Stock Price Index Number. Journal of the American Statistical Association. 33. 10.1080/01621459.1938.10502333.
20. Sohn, Kyoung-Woo & Kim, Sang-Su. (2016). A Study on the Predictive Power of the Stock Price Index. Journal of Money & Finance. 30. 95-122. 10.21023/JMF.30.1.4.
21. Neusser, Klaus. (2016). Autoregressive Moving-Average Models. 10.1007/978-3-319-32862-1_2.
22. Zhang, Zhihua& Moore, John. (2015). Autoregressive Moving Average Models. 10.1016/B978-0-12-800066-3.00008-5.
23. Fabozzi, Frank &Focardi, Sergio &Rachev, Teodosii&Arshanapalli, Bala. (2014). Autoregressive Moving Average Models. 10.1002/9781118856406.ch9.
24. Akuffo, Buckman&MintahAmpaw, Enock. (2020). An Autoregressive Integrated Moving Average (ARIMA) Model For Ghana's Inflation (1985 -2011).
25. Zainorzuli, Siti&Che Abdullah, Syahrul Afzal & Adnan, Ramli&AhmatRuslan, Fazlina. (2019). Comparative Study of Elman Neural Network (ENN) and Neural Network Autoregressive With Exogenous Input (NARX) For Flood Forecasting. 11-15. 10.1109/ISCAIE.2019.8743796.