# NEURAL NETWORKS FOR TIME SERIES FORECASTING TO PREDICT THE RETURN OF STOCK INDEX

**T.Kamalavalli[1] , S.Naveen Dhinakaran [2]**

[1]Assistant Professor,

Department of Computer Applications,

PSNA College of Engineering and Technology, Dindigul

[2]S.Naveen Dhinakaran

## Abstract

*Generating accurate P**redictions of Financial Market** movements has always been a key goal for all stakeholders of an economy. The various algorithms used for time series forecasting. It can be categorized into linear model (ARIMA, SARIMA) and non-linear models (neural networks). A **linear regression model** is not suitable for time series because these model are neglected a problems on time. It adds additional information that makes time series problems even more difficult to handle .The **traditional model** used to predict time series like autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average model (SARIMA) but this model is not sufficient for time series .that's why Neural Networks (NN) is important essential for time series forecasting. Only certain type of N**eural Networks Suitable for time series model** .Such as Convolutional neural network (CNN) and Long Short-Term Memory (LSTM) neural networks. These machine learning models that enable better predictions by improving the accuracy of time series. In this paper , different time series models are taken and compare them with neural networks for predict the returns for the S&P500 (Standard & Poor's 500 ) index .This paper also concludes that neural networks is suited for time series forecasting . The experiments were executed in jupyter notebook 6.0.3 .the codes and simulations were implemented using python 3.6.5 with a dependency of tensorflow 1.15.0 and keras 2.3.1*

**Keywords: Neural Networks - Time Series Forecasting - The S&P500 Index - Returns of Stock Index**

## 1. Introduction

The S**tock Market** refers to the collection of markets and exchanges where regular activities of buying, selling, and issuance of shares of publicly-held companies take place. When we talk about Indian **Stock Exchanges**, most of the investing population has heard of two stock exchanges in India. Bombay Stock Exchange (BSE) and National Stock Exchange (NSE). A S**tock Exchange** is a place or platform which hosts a market where buyers and sellers come together to trade stocks during specific hours of business days. A **stock or share** also known as a company's equity is a financial instrument that represents ownership in a company or corporation and represents a proportionate claim on its assets what it owns and earnings what it generates in profits. The **share prices** are set by supply and demand in the market as buyers and sellers place orders. An **index** measures the price of stock and performance of a basket of securities using a standardized metric and methodology. A passive index investing has become a popular low-cost way to replicate the returns of popular indices such as the S&P 500. The S**&P 500 indexes** or the Standard & Poor's 500 index is a market-capitalization-weighted index of the 500 largest U.S publicly traded companies. The S&P is a float-weighted index, meaning company market capitalizations are adjusted by the number of shares available for public trading. As a result, there are many funds designed to track the performance of the S&P. T**ime Series Forecasting** is a technique for the prediction of events through a sequence of time. The techniques predict future events by analyzing the trends of the past on the assumption that future trends will hold similar to historical trends. Time series forecasting starts with a historical time series. It analysts examine the historical data and check for patterns of time decomposition. Neural Networks generates the best possible result without needing to redesign the output criteria. Neural network helps to predict time series forecasting**.** Because its easy-to-extract features, good at extracting patterns easily, support for multiple inputs and outputs, easy to predict from training data in **Financial Market.**

## 2. Literature survey

In this research study, the following models are taken for literature study and are compared with proposed models.

**2.1. Linear Regression**- It is a Statistical Method for examining the relationship between a dependent variable and one or More Independent Variables. [13]. The problem with Linear Regression, It's a dependent variable usually affected by the independent variables. Linear Regression is limited to linear relationships. It only looks at the mean of the dependent variable and also linear regression is sensitive to outlier.

**2.2. ARIMA -**Based on Autoregressive Integrated Moving Average, or ARIMA, is a forecasting method for univariate time series data [3]. It supports both an autoregressive and moving average elements. The integrated element refers to differencing allowing the method to support time series data with a trend. A Problem with ARIMA is that it does not support seasonal data. That is a time series with a repeating cycle. ARIMA expects data that is either not seasonal or has the seasonal component removed [5].

**2.3. SARIMA -**Based on Seasonal Autoregressive Integrated Moving Average, or SARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component [14]. The problem with SARIMA model is, it can only extract linear relationships within the time series data.

**2.4. Neural Network -**A Neural Network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Neural Networks are potentially useful endemic time series forecasting methods because of their strong nonlinear mapping ability and tolerance to complexity in forecasting data. They are especially useful when a nonlinear relationship exists within the time series data.

## 3. Related Work

Before deep learning era financial time series modeling has mainly concentrated in the linear model. Any modifications on this model and the result have proved that the traditional time series model does provide decent predictive power to a limit forecasting. The work showed that provided more accurate forecasts than the back-propagation of a neural network [11][9]. Recently deep learning methods have demonstrated better performances when compare to linear model. Because of the improved computational power and the ability of learning non-linear relationships enclosed in various financial features [6].

Compared different deep learning architectures including LSTM and CNN models for the prediction of stocks index they concluded that CNN architecture is capable of identifying changes in trend of stocks and outperforms other models. LSTM and showed that the resulting model beat the performance of traditional model. It also improves the accuracy of the prediction. The performance of LSTM and CNN will be further boosted by feeding relevant data based on financial domain knowledge they provide better result with much lower prediction errors [3].

## 4. Proposed Work

### 4.1. Data and Methodology

This analysis involves information on the stock indices of Nifty 500 Index covering the period from 2000 to 2020 and having a total number of 7661 observations. On the idea of this information, to determine various time series model to predict the returns for the S&P 500 index.

The proposed work focus on EDA (**Exploratory Data Analysis).** EDA Process Specifically Focuses on Understanding Characters of Data, Finding Meaningful Patters in Data, Possible Modeling Strategies, Debugging Strategies and Visualization of Results.

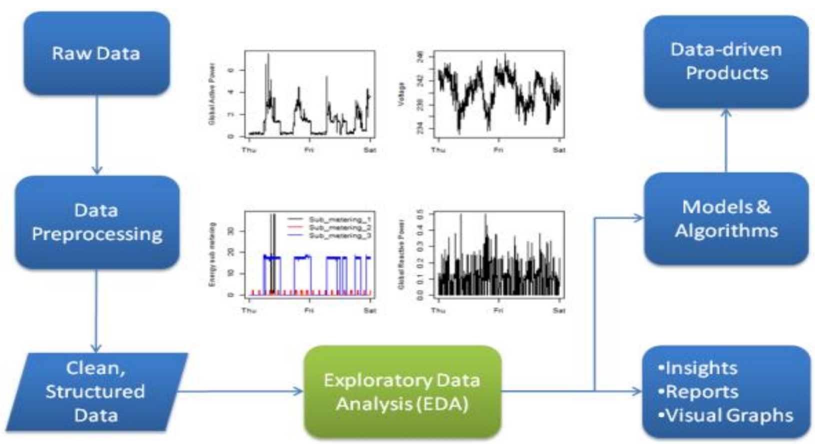



**Figure 1: Exploratory Data Analysis**

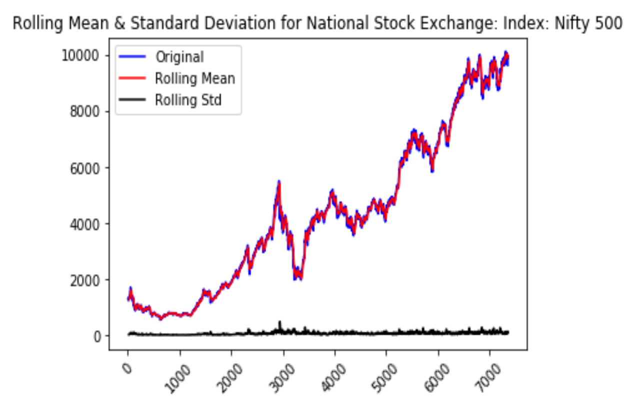
**Table 1: Nifty 500 Index Data Set**

| | date | Bombay Stock Exchange: Index: 100 | Bombay Stock Exchange: Index: SENSEX | Bombay Stock Exchange: Index: 200 | Bombay Stock Exchange: Index: Dollex-200 | Bombay Stock Exchange: Index: 500 | Bombay Stock Exchange: Index: Information Technology | Bombay Stock Exchange: Index: Capital Goods | Bombay Stock Exchange: Index: Fast Moving Consumer Goods | Bombay Stock Exchange: Index: Consumer Durables | ... | National Stock Exchange: Index: Nifty 500 | National Stock Exchange: PE Ratio: Nifty 50 | National Stock Exchange: PB Ratio: Nifty 50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2000-01-03 | 1636.49 | 5375.11 | 635.04 | 243.17 | 1920.08 | 4260.20 | 1196.46 | 1172.50 | 1526.33 | ... | 1291.55 | 25.91 | 4.63 |
| 3 | 2000-01-04 | 1690.67 | 5491.01 | 652.91 | 249.89 | 1972.29 | 4554.08 | 1179.47 | 1138.74 | 1517.65 | ... | 1335.45 | 26.67 | 4.76 |
| 4 | 2000-01-05 | 1643.14 | 5357.00 | 636.89 | 243.65 | 1922.87 | 4342.29 | 1145.40 | 1116.22 | 1470.10 | ... | 1303.80 | 25.97 | 4.64 |
| 5 | 2000-01-06 | 1644.78 | 5421.53 | 636.96 | 243.68 | 1923.36 | 4160.10 | 1174.28 | 1168.77 | 1531.15 | ... | 1306.60 | 26.32 | 4.70 |
| 6 | 2000-01-07 | 1606.46 | 5414.48 | 620.91 | 237.59 | 1871.82 | 3828.79 | 1138.47 | 1207.37 | 1497.40 | ... | 1276.30 | 26.25 | 4.69 |
| | | | | | | | | | | | | | | |
| 7359 | 2020-02-24 | 11941.07 | 40363.23 | 4989.47 | 1156.22 | 15469.29 | 16176.41 | 16374.24 | 11308.68 | 26874.39 | ... | 9758.20 | 26.92 | 3.33 |
| 7360 | 2020-02-25 | 11903.01 | 40281.20 | 4972.74 | 1151.86 | 15418.71 | 16270.52 | 16262.28 | 11306.25 | 26572.20 | ... | 9724.70 | 26.85 | 3.32 |
| 7361 | 2020-02-26 | 11777.28 | 39888.96 | 4919.40 | 1143.14 | 15256.99 | 16022.85 | 15921.25 | 11238.48 | 26442.94 | ... | 9622.70 | 26.58 | 3.29 |

5258 rows × 61 columns

## 4.2. Data Stationary

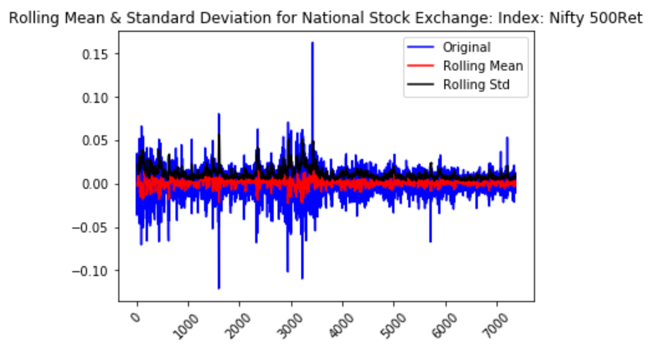
It's a property of time series data stating that the distributional properties (mean and standard deviation) of the data series have not changed across time for forecasting. It's important that the data be stationary because in the absence of stationary, one is asking the model to predict data that is nothing like anything it has seen before. A Common Test of Stationary is The Dickey-Fuller Test. Such as if the P-Value associated with the Dickey-Fuller Test Statistic is Greater than 0.05, State that the Data Is Not Stationary.




```
Results of Dickey-Fuller Test:
Test Statistic                    0.078353
p-value                           0.964552
# Lags Used                      13.000000
Number of Observations Used    5244.000000
Critical Value (1%)              -3.431598
Critical Value (5%)              -2.862091
Critical Value (10%)             -2.567063
dtype: float64
```

**Figure 2: Without Data Stationary of Nifty 500 index**

From the above results, the p-value is 0.964552; leading to the conclusion that the nifty500 index series is not stationary. It can be further seen more clearly from the graph. The clear upward trend of the rolling mean and rolling standard deviation signals that nifty500 is not stationary.To make the data as stationary as possible the easiest ways of making the data stationary is to calculate the first difference for level variables (like indices) .the first difference in the percentage change in relation to the previous time period. For percentage variables, it’s the value of the variable in the current time period subtracted from the value of the variables at the previous date.




```
Results of Dickey-Fuller Test:
Test Statistic                -1.741829e+01
p-value                        4.842084e-30
# Lags Used                    1.300000e+01
Number of Observations Used    5.243000e+03
Critical Value (1%)           -3.431598e+00
Critical Value (5%)           -2.862091e+00
Critical Value (10%)          -2.567064e+00
dtype: float64
```

**Figure 3: With Data Stationary for Nifty 500 index**

From the above graph, the percentage change in nifty 500 indexes is stationary as the p-value is less that 0.05.again, for greater intuition look at the graph - the rolling mean and rolling standard deviation lines are along are almost flat meaning that they don't change in a statistically significant manner over time.

## 4.3. Training, Validation and Testing

The objective of a machine learning methods is to predict the dependent variable as accurately as possible. To minimize the predicted error (the difference between the actual value and predicted value).to measure the predictive accuracy of a model, it is important that the forecast accuracy be measured out-of sample as the training accuracy can be made arbitrarily high through over fitting. If we use the entire out-of-sample data for testing, we may over fit to the out-of-sample data (data leakage), resulting poor true generalizability.To protect against ‘data leakage’, split the out-of-sample data into two parts: validation data and testing data. The training, validation and testing data can be organized in many ways such as,

**Cross Validation**: Bootstrap sampling for cross sectional methods but not suited for time series.

**Fixed Window**: Training, Validation and Testing Periods Demarcated By Dates

**Rolling Window**: Shifting a Window of Fixed Size - By One Observation Successively Expanding Window - Increasing the Window Size by 1 Successively

### 4.4. Rescaling Normalized Data

Normalize the data in the training sample because machine learning methods are not scale invariant. Scale it to being between -1 and 1 as that is the appropriate scaling for data when being input into a Convolutional Neural Network (CNN) And Long Short Term Memory Network (LSTM).Here scaling for all the methods to make the prediction errors comparable.

### 4.5. Metric to be used to Gauge Accuracy of Forecasts

The most commonly used metric for gauging the accuracy of time series forecasts is the **Mean Squared Error (MSE).**

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \quad \textbf{(1)}$$

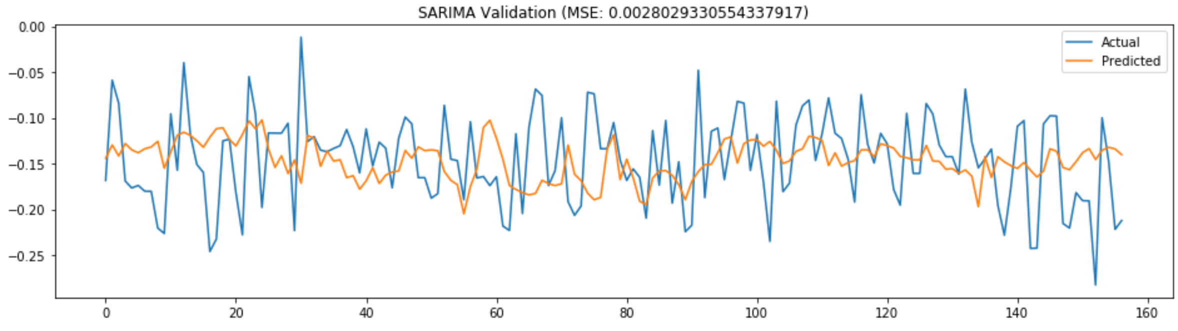
The higher the MSE, the worse is the predictive accuracy of the model. Thus for each model, it is desired that the MSE be minimized.

## 5. Experimental Results

After training the model, the result of testing has shown different results. The proposed work uses data from the past to predict the return on the dataset. To assess the model, our primary model performance metric is Mean Squared Error (MSE). The MSE will be performed on the test data, which is totally unseen from the training and development stages. It is calculated between our predicted price and the true price.

### 5.1 Linear Model - ARIMA / SARIMA model

ARIMA models describe how each successive observation is related to the previous observation [6]. The seasonal **ARIMA (SARIMA**) is capable of modeling the seasonal components in a univariate time series in addition to the autoregressive, moving average and trend components typically modeled by ARIMA.



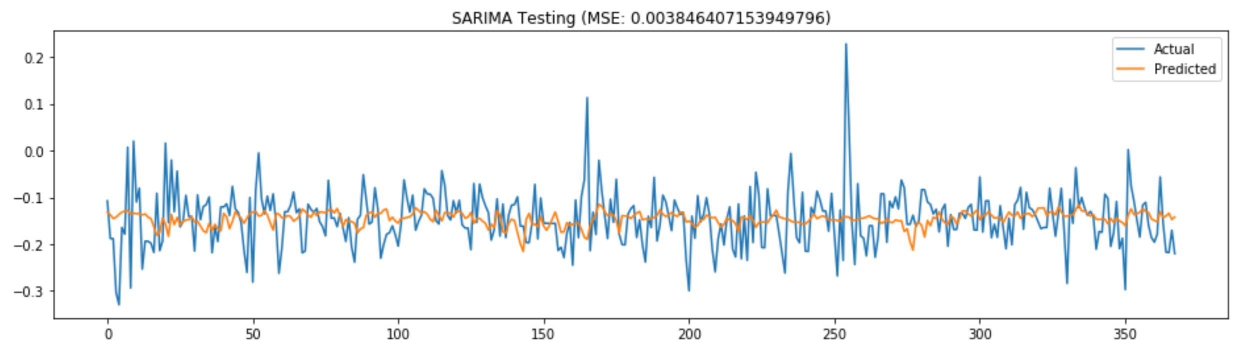



**Figure 4: Result of ARIMA / SARIMA MODEL**

Based on above results we face challenges on ARIMA models. They are generally poor at predicting turning points. As the number of steps forecasted ahead in the future increases, the forecast converges to the mean. The model of ARIMA and SARIMA, where no differencing was performed to change the stationarity. However, the model gave a linear prediction from test data, even the forecasted trend is opposite to the real price trend. The results suggested ARIMA model did not perform well in predicting non-linearity and long-term prediction.

## 5.2 Neural Network Model

A neural networks are forecasting methods that are based on simple mathematical models of the brain.

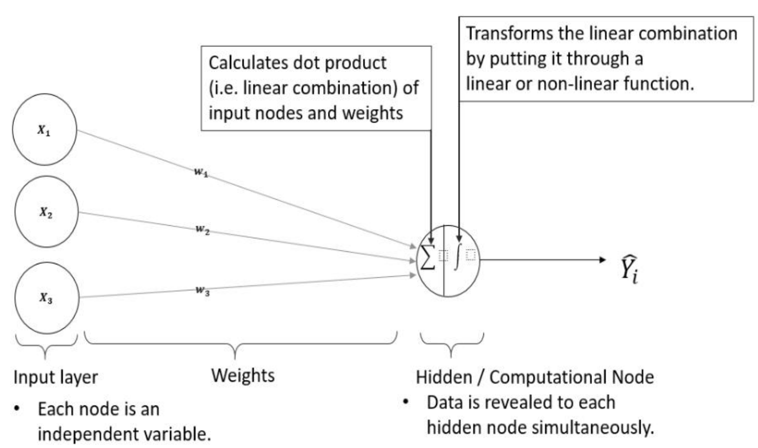



**Figure 4: NEURAL NETWORK MODEL**

Here We Consider Two Methods:

1) 1Dimensional Convolutional Neural Networks (**1D-CNNs**)

2) Long Short Term Memory (**LSTM**) Networks

### 5.2.1. Convolutional Neural Network (CNN) Model:

Convolutional Neural Networks (CNNs) can extract information from the temporal structure of the data by preserving the spatial/ temporal structure of the data in the input layer. Using filters which look for patterns in spatially adjacent data. The information extracted by the filters is known as a Feature Map.Each additional feature results in a new feature map extracting a more complex feature. For a time series dataset, a filter can only move along 1 dimension - time

The CNN Layer may be followed by a sub-sampling layer which reduces the noise in the learned features (i.e. the feature maps). The Sub-Sampling Layer is followed by a Regression Layer .

CNN Model Training Start:

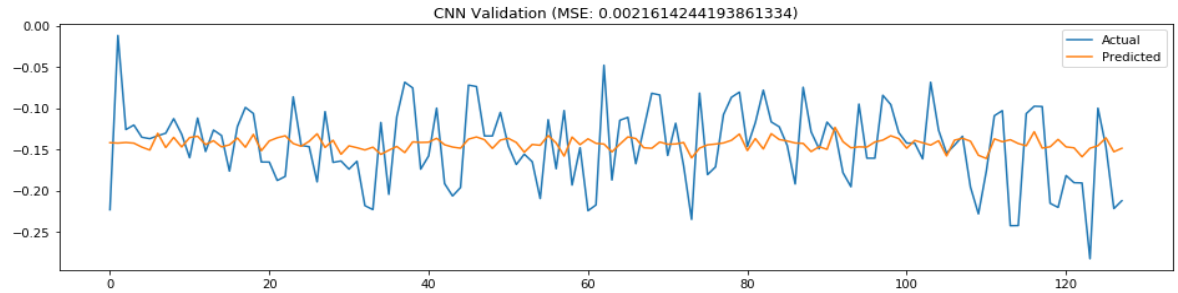
```
Train on 4693 samples, validate on 128 samples
Epoch 1/100000
 - 1s - loss: 0.1225 - val_loss: 0.0313
Epoch 2/100000
 - 1s - loss: 0.0532 - val_loss: 0.0297
Epoch 3/100000
 - 1s - loss: 0.0473 - val_loss: 0.0278
Epoch 4/100000
 - 1s - loss: 0.0439 - val_loss: 0.0242
Epoch 5/100000
 - 1s - loss: 0.0407 - val_loss: 0.0250
Epoch 6/100000
 - 1s - loss: 0.0388 - val_loss: 0.0222
Epoch 7/100000
 - 1s - loss: 0.0361 - val_loss: 0.0212
Epoch 8/100000
 - 1s - loss: 0.0349 - val_loss: 0.0208
```

CNN Model Training End:

```
Epoch 420/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 421/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 422/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 423/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 424/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 425/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 426/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
Epoch 427/100000
 - 1s - loss: 0.0096 - val_loss: 0.0023
```

### CNN Training Results:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 15, 5)             6005
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 15, 5)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 15, 5)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 75)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 76
=================================================================
Total params: 6,081
Trainable params: 6,081
Non-trainable params: 0
```



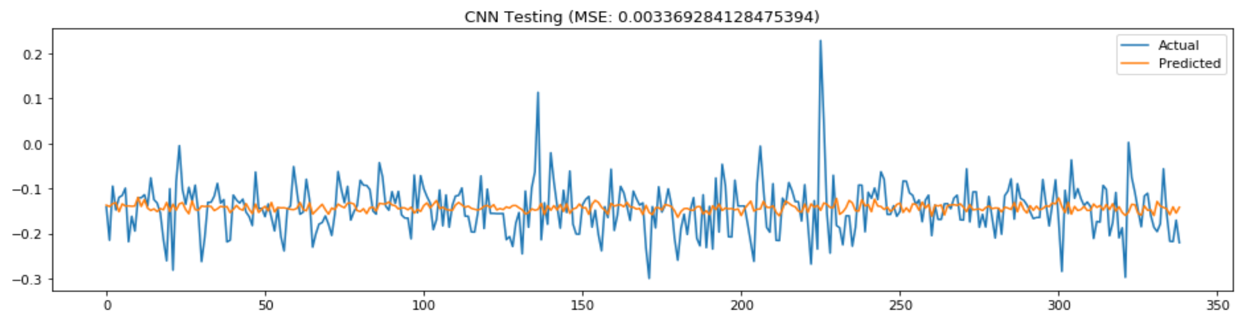




**Figure 6: Result of CNN MODEL**

Based on the above results the MSE is reduced compared to ARIMA/SARIMA model.

### 5.2.2. Long Short Term Memory Network (LSTM)

A CNNS for time series data is that they do not draw information from the sequential nature of independent data. LSTM networks draw information from the sequential nature of the data because they have memory and it remembers what it has seen. LSTMS build short and long term memories by revealing data to the hidden nodes in a sequential.Each LSTM node is comprised of LSTM cells. The long term and short term memory is updated in each LSTM cell upon being exposed to each subsequent element of the sequence, conditional on the output of each 'gate'. Gate is a neural network with a sigmoid/logistic activation function which determines how the memory updated by accepting as inputs as the current elements x of the sequence and the outputs of the previous LSTM cell.

LSTM Model Training Start:

```
[0.5488135  0.71518937 0.60276338 0.54488318]
Train on 4693 samples, validate on 128 samples
Epoch 1/100000
 - 20s - loss: 0.3244 - val_loss: 0.0025
Epoch 2/100000
 - 20s - loss: 0.0280 - val_loss: 0.0027
Epoch 3/100000
 - 22s - loss: 0.0254 - val_loss: 0.0027
Epoch 4/100000
 - 23s - loss: 0.0240 - val_loss: 0.0028
Epoch 5/100000
 - 23s - loss: 0.0235 - val_loss: 0.0027
Epoch 6/100000
 - 22s - loss: 0.0225 - val_loss: 0.0025
Epoch 7/100000
 - 22s - loss: 0.0223 - val_loss: 0.0026
Epoch 8/100000
 - 23s - loss: 0.0214 - val_loss: 0.0030
Epoch 9/100000
```

LSTM Model Training End:

```
Epoch 149/100000
 - 22s - loss: 0.0100 - val_loss: 0.0022
Epoch 150/100000
 - 22s - loss: 0.0100 - val_loss: 0.0022
Epoch 151/100000
 - 22s - loss: 0.0098 - val_loss: 0.0022
Epoch 152/100000
 - 22s - loss: 0.0099 - val_loss: 0.0022
Epoch 153/100000
 - 22s - loss: 0.0099 - val_loss: 0.0022
Epoch 154/100000
 - 22s - loss: 0.0100 - val_loss: 0.0022
Epoch 155/100000
 - 22s - loss: 0.0100 - val_loss: 0.0022
Epoch 156/100000
 - 22s - loss: 0.0100 - val_loss: 0.0022
```

## Training Result of LSTM:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 300)               1081200
_________________________________________________________________
dropout_2 (Dropout)          (None, 300)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 301
=================================================================
Total params: 1,081,501
Trainable params: 1,081,501
Non-trainable params: 0
_________________________________________________________________
```

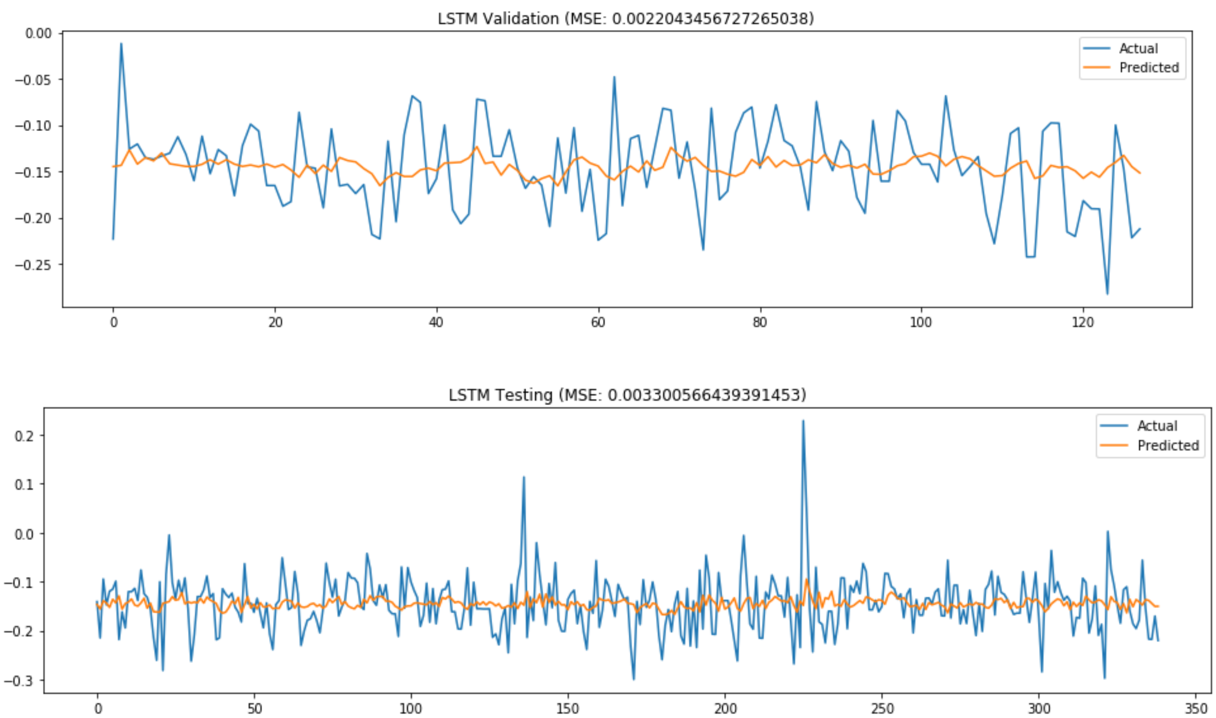



**Figure 7: Result of LSTM MODEL**

**Table 2: Performance Analysis**

| Name of the Model | Type of Relation | Accuracy of MSE (In terms of %) | |
|---|---|---|---|
| | | Result of Validation | Result of Test |
| ARIMA/ SARIMA | Linear | 0.28 | 0.38 |
| CNN | Non Linear | 0.21 | 0.33 |
| LSTM | Non Linear | 0.22 | 0.33 |

In Table 2, shows that each model was trained the index of the stock and the MSE for the testing sample are performed on each of stocks. Comparison of the results from each model predicts that **CNN** and **LSTM** perform better than both traditional models as expected. As for long-term time series prediction, the **Neural Networks** brings advantages of selecting the important and relevant information hence enhancing the predictive accuracy. The return of our trading strategy is the sum of the individual returns from each of the stocks above. The benchmark return is used for comparing the performance of the strategies and annual return of S&P 500.

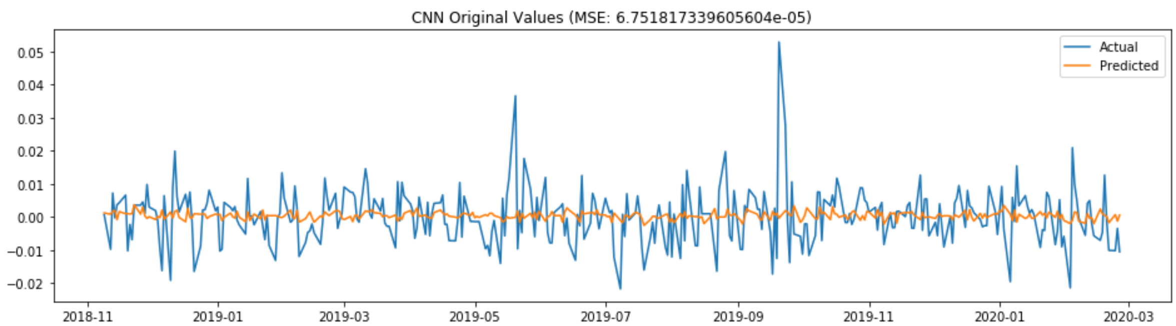



**Figure 8: Neural Network Result S&P 500 Index**

## 6. Conclution

In proposed system, NEURAL NETWORK is used for predicting the stock price. Here, trained the model for predict the stock index price from the obtained results. it is quite obvious that these models are reasonably efficient in recognizing the patterns that exists in the stock market. This shows that there is an underlying dynamics, which is very common to all the stock markets. The linear series models such as ARIMA/SARIMA time-series based prediction models and they are incapable in recognizing the under dynamics with the help of multivariate time series. Hence from the results obtained, Neural Networks proved as a better performer when compared to other model by implemented CNN and LSTM Neural Networks for time-series prediction problems. CNN and LSTM neural networks are best suited for time series forecasting. The deep learning model might possibly apply in other areas as well as in the future.

## Reference

[1] M.R. Amer, B. Siddiquie, C. Richey, and A. Divakaran, "Emotion detection in speech using deep networks.", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) , pp. 3724-3728, IEEE, 2014.

[2] Y. Bengio, "Learning deep architectures for AI. Foundations and trends in Machine Learning“

vol. 2, no. 1, pp. 1-127, 2009

[3] Bijesh Dhyani, Manish Kumar, Poonam Verma, Abhisheik Jain, “Stock Market Forecasting Technique using Arima “International Journal of Recent Technology and Engineering (IJRTE) ,ISSN: 2277-3878, Volume-8 Issue-6, March 2020

[4] G.E. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung, " Time series analysis: forecasting and control. ", Holden-Day, 1976.

[5] B. Choi, "ARMA model identification", Springer Science & Business Media, 2012.

[6] Hiransha M , Gopalakrishnan E,”NSE Stock Market Prediction Using Deep-Learning Models” ScienceDirect, Procedia Computer Science 132 (2018) 1351–1362

[7] G.E. Hinton, et al, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97, 2012.

[8] J.J.M. Moreno, A.P. Pol, and P.M. Gracia, "Artificial neural networks applied to forecasting time series. ", Psicothema, vol. 23, no. 2, pp. 322-329. 2011.

[9] Ramaswamy Seethalakshmi,”Analysis of stock market predictor variables using Linear Regression”, International Journal of Pure and Applied Mathematics, January 2018,119(15):369-377

[10] R. Salakhutdinov, A. Mnih, and G.E Hinton, " Restricted Boltzmann machines for collaborative filtering. " In Proceedings of the 24th international conference on Machine learning, pp. 791-798. ACM. 2007.

[11] G.P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model.", Neuro computing, vol. 50, pp. 159-175, 2003.

[12] G.P, Zhang, "Neural networks for time-series forecasting. ", In Handbook of Natural Computing, Springer Berlin Heidelberg, pp. 461-477, 2012.

[13] https://sciencing.com/advantages-using-independent-group-ttest-8647277.html

[14] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3641111