

QATAR UNIVERSITY

COLLEGE OF ENGINEERING

STOCK MARKET FORECASTING: AN APPLICATION OF LONG SHORT TERM

MEMORY (LSTM) RECURRENT NEURAL NETWORK

BY

HISHAM YASSIN

A Project Submitted to  
the Faculty of the College of  
Engineering  
in Partial Fulfillment  
of the Requirements  
for the Degree of  
Masters of Science in Computing

January 2018

© 2018 Hisham Yassin. All Rights Reserved.

## COMMITTEE PAGE

The members of the Committee approve the Project of Hisham Yassin  
defended on 24/12/2017.

---

Dr. Abdelaziz Bouras and Dr. Ali Jaoua  
Thesis/Dissertation Supervisor

---

Dr. Sumaya Al-Maadeed  
Committee Member

---

Dr. Khaled Shaban  
Committee Member

---

Dr. Jihad Al-Jaam  
Committee Member

## ABSTRACT

Hisham Yassin, Masters: January : 2018, Masters of Science in Computing

Title: Stock Market Forecasting: An Application of Long Short Term Memory (LSTM) Recurrent Neural Network

Supervisor of Project: Dr. Abdelaziz Bouras and Dr. Ali Jaoua

Predicting stock market prices is regarded as a challenging task of financial time series, due to its chaotic, non-linear, non-stationary and dynamic nature. In this project we address the problem of stock market forecasting by making a comparison between different machine learning prediction models mainly Support Vector Machine (SVM), Artificial Neural Networks (ANN), Random Forest (RS), and Long Short Term Memory (LSTM) Recurrent Neural Network. For this goal, different models are built for predicting stock prices for 10 days in advance, and a number of experiments were executed based on ten years of historical data for stock prices from different sectors of the industry of the Qatari and the American markets. The results were analyzed using Mean Squared Error (MSE) and Mean Absolute Error (MAE) measuring metrics.

Furthermore, we developed an application for predicting stock prices and trend movement with a motivation that trading strategies and investment decisions are more reliable and efficient when guided by forecasts which could lead to more profit.

## DEDICATION

*To my parents:*

*Omar Yassin and Nadia Fattah, who have encouraged me to pursue this degree and who have always been my role model*

*To my dear wife and kids:*

*Samar, Khalid and Wael, for your endless love and support through the difficult times*

## ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor *Prof. Abdelaziz Bouras* and my co-supervisor *Prof. Ali Jaoua*. This project would not have been done without your sincere support. Your patience, advice and motivation have always guided me to the right direction, and helped me overcome a lot of difficulties. It was an honor to work with you.

From the bottom of my heart, I say thank you.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	V
LIST OF ABBREVIATIONS .....	VIII
LIST OF TABLES .....	IX
LIST OF FIGURES .....	X
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Goals and Objectives:.....	3
<b>CHAPTER 2: BACKGROUND AND RELATED WORK .....</b>	<b>4</b>
2.1 Stock Market Technical Analysis: .....	4
2.2 Machine Learning: .....	6
2.2.1 Support Vector Machine (SVM) .....	6
2.2.2 Artificial Neural Networks (ANN).....	8
2.2.3 Long Short Term Memory (LSTM) .....	10
2.3 State of the art .....	14
<b>CHAPTER 3: METHODOLOGY AND VALIDATION.....</b>	<b>16</b>
3.1 Data Collection.....	16
3.2 Preprocessing .....	19
3.2.1 Lag and Sequences Time Series Format.....	19
3.2.2 Data Normalization .....	20

3.3 Experiments.....	21
3.3.1 Overview .....	21
3.3.2 Experiment Baseline.....	26
3.3.3 Results .....	27
3.3.4 Discussion.....	31
<b>CHAPTER 4: CASE STUDY – DAILY STOCK PREDICTION APPLICATION.</b>	<b>33</b>
4.1 System Overview .....	33
4.2 System Applicability and Usability.....	38
4.3 System Limitations.....	38
<b>CHAPTER 5: CONCLUSIONS AND FUTURE WORK.....</b>	<b>40</b>
5.1 Conclusion.....	40
5.2 Future Work Directions.....	41
<b>REFERENCES.....</b>	<b>43</b>
<b>APPENDIX.....</b>	<b>46</b>
APPENDIX A: QATAR EXCHANGE INDEX (DSM) .....	46
APPENDIX B: BOMBAY STOCK EXCHANGE SENSITIVE INDEX (BSESN) .....	48

## LIST OF ABBREVIATIONS

- ANN: Artificial Neural Network
- CCI: Commodity Channel Index
- EMA: Exponential Moving Average
- LSTM: Long Short-Term Memory
- MACD: Moving Average Convergence/Divergence
- MAE: Mean Absolute Error
- MLP: Multi-Layer Perceptron
- MSE: Mean Squared Error
- NLP: Neutral Language Processing
- RF: Random Forest
- RMSE: Root Mean Squared Error
- RNN: Recurrent Neural Network
- RSI: Relative Strength Index
- SMA: Simple Moving Average
- SVM: Support Vector Machine
- SVR: Support Vector Regression
- WMA: Weighted Moving Average



## LIST OF TABLES

Table 1: Models Configuration .....	24
Table 2: MAE Results for Prediction Models and the Baseline .....	27
Table 3: MSE Results for Prediction Models and the Baseline .....	28
Table 4: RMSE Results for Prediction Models and the Baseline .....	28
Table 5: LSTM Evaluation Results of DSM Dataset.....	46

## LIST OF FIGURES

Figure 1: SVM Kernels .....	7
Figure 2: The Perceptron – An Artificial Neuron .....	8
Figure 3: Typical Artificial Neural Network – Multi-layer Perceptron.....	9
Figure 4: Recurrent Neural Network .....	10
Figure 5: RNN Architecture (left) vs LSTM Architecture (right) .....	13
Figure 6: Historical Closing Prices for Ooredoo (ORDS.QA), Qatar National Bank (QNBK.QA) and Qatar Exchange Index (DSM) .....	17
Figure 7: Historical Closing Prices for Alphabet Inc. (GOOGL), Microsoft Corporation (MSFT), Apple Inc. (AAPL) and Amazon.com, Inc. (AMZN).....	18
Figure 8: S&P Bombay Stock Exchange Sensitive Index (^BSESN) .....	18
Figure 9: Data before Transformation – AAPL Dataset.....	20
Figure 10: Lag and Seq transformation for Closing prices - AAPL Dataset .....	20
Figure 11: Methodology Overview.....	23
Figure 12: LSTM Model Architecture using Keras .....	25
Figure 13: Two Stage Hybrid Prediction Model of Patel et al., 2015 [4].....	26
Figure 14: Linear SVR.....	28
Figure 15: LSTM Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	29
Figure 16: LinearSVR Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	29
Figure 17: MLP Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10”	

(right) Days .....	29
Figure 18: RN model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	30
Figure 19: SVR Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	30
Figure 20: SVR-ply Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	30
Figure 21: SVR-RBF Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	31
Figure 22: SVR-RBF-2 Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days .....	31
Figure 23: System Overview – Daily Stock Prediction Application .....	34
Figure 24: Trending Stocks List .....	36
Figure 25: 10 Days Price Forecasts and Trending Direction .....	37
Figure 26: DSM Actual Prices vs Forecasted Prices for 10 days forecasts .....	47

## **CHAPTER 1: INTRODUCTION**

Predicting stock market prices is regarded as a challenging task of financial time series. For years, investors have accepted the Efficient Market Hypothesis (EMH) for claiming that future stock prices are partially predictable. This is due to the fact that existing share prices reflect all the relevant information required to predict a stock price based of historical trading data [1]. There is also the Random-Walk hypothesis which states that stock market prices are essentially random and any attempt to predict the stock prices will fail. [2] Although predictable, based on the Efficient Market hypothesis, but it remains hard to predict stock prices “since the stock market is essentially dynamic, nonlinear, complicated, nonparametric, and chaotic in nature” [3]. There are many additional factors that cause fluctuation of stock prices such as political situations, company performance, economic activities and other unexpected events. These factors make the stock movement nonlinear, uncertain, and non-stationary. Hence, it is quite difficult to predict stock market prices and their movement. For this reason, investors usually use technical analysis for evaluating stocks by studying the statistics generated from the market activity mainly past prices and volumes. These statistics are presented in the form of charts to find patterns that might suggest future stock behavior. In addition to that, investors pay a close attention to recent news in order to avoid investing in risky stocks and to make correct stock investment decisions [1].

Over the years, a number of applications have shown supervised machine learning models such as Genetic Algorithms, Support Vector Machine, Artificial Neural Network and Random Forests can be useful tools to predict the movement trend of stock prices, due

to their ability to handle non-linear systems. However most of them have still not given satisfactory results with very high accuracy and stable performance on stock prediction.

Recently Recurrent Neural network (RNN) has been receiving great attention for modeling time series and sequence problems, due to internal cyclic connections between units that can store information while processing new inputs. However, RNN suffers from the Vanishing Gradient Problem, since RNN are trained by back propagation through time. When the gradient is passed back through many time steps, it tends to grow or vanishes. This problem makes it difficult for RNN models to learn long-range dependencies. Long Short-Term Memory (LSTM) is a special variant of RNN that solves the problem of vanishing gradient by incorporating a Memory Cell with three main gates: a forget gate, an input gate, and an output gate. With this architecture, not only an LSTM network is able to remember information for much longer periods of time, it is able to remember what is important and how long it will remain important through maintaining the state of the memory cell.

The remainder of this report is organized as follows: Section 1.1 describes the problem statement and the objectives addressed in the project. Chapter 2 presents a background overview about stock market forecasting and related works from the literature review. Chapter 3 describes the methodology followed for tackling the problem, experiments and evaluation results are also presented and discussed in this chapter. Chapter 4 presents the Daily Stock Prediction (DSP) web application, and finally, we conclude in Chapter 5 with a brief summary of the achievements and a discussion of future works.

## **1.1 Goals and Objectives:**

The objective of this project is to study the applicability of recurrent neural networks, in particular the LSTM networks on the problem of stocks market prices prediction, and compare their performance with traditional machine learning prediction models as well as with a baseline model developed by *Patel et al., 2015* [4]. Also we will develop a stock forecasting web application that will predict stock prices for ten days in the future using LSTM model as the core prediction engine. The motivation behind developing this forecasting application is to give users more confidence in making investment decisions and develop trading strategies based on forecasted future prices and future trend movement of a stock predicted by a reliable model such as the LSTM.

## **CHAPTER 2: BACKGROUND AND RELATED WORK**

In this section, existing academic literature about predicting stock market prices and trend movement using machine learning or deep learning techniques with technical analysis for improving the efficiency will be reviewed. In section 2.1, we will take a look at technical analysis and how it is used. In section 2.2, we will explore the most used machine learning algorithms in the field of stock market forecasting. Section 2.3, talks about the recent research work done using state-of-the-art machine learning techniques.

### **2.1 Stock Market Technical Analysis:**

Technical analysis is a popular approach used by market analysts and professional traders by making use of recurring patterns and trends within the historical prices of a stock. Technical analysis studies the statistics generated by the market activities mainly past prices and volume, it employs the use of sophisticated charts to identify patterns and trends in order to predict future prices or movement of a stock.

In order to develop trading strategies analysts have used a number of technical indicators. The most common technical indicators used in the academic literature are shown below with their mathematical formulas:

$$\text{Simple Moving Average (SMA)} = \frac{\sum_1^n \text{Close}}{n}$$

$$\text{Weighted Moving Average (WMA)} =$$

$$\frac{n * \text{Close}_t + (n-1) * \text{Close}_{t-1} + (n-2) * \text{Close}_{t-2} \dots + (1) * \text{Close}_{t-(n-1)}}{n!}$$

$$\text{Momentum} = \text{Close}_t - \text{Close}_{t-1}$$

$$\text{Stochastic K\%} = \frac{\text{Close} - \text{LowestLow}_{[\text{last } n \text{ periods}]}}{\text{HighestHigh}_{[\text{last } n \text{ periods}]} - \text{LowestLow}_{[\text{last } n \text{ periods}]}} * 100$$

$$\text{Stochastic D\%} = \text{SMA} (\text{Stochastic K\%})$$

$$\text{Larry William's R\%} = \frac{\text{HighestHigh}_{[\text{last } n \text{ periods}]} - \text{Close}}{\text{HighestHigh}_{[\text{last } n \text{ periods}]} - \text{LowestLow}_{[\text{last } n \text{ periods}]}} * 100$$

$$\text{Relative strength index (RSI)} = 100 - \frac{100}{1 + (\sum_{i=0}^{n-1} \text{UP}_{t-i}/n) / (\sum_{i=0}^{n-1} \text{DW}_{t-i}/n)}$$

$$\text{Moving Average Convergence Divergence (MACD)} = \text{MACD}(n)_{t-1} + \frac{2}{n+1} *$$

$$(\text{DIFF}_t - \text{MACD}(n)_{t-1})$$

$$\text{A/D (Accumulation/Distribution) Oscillator} = \frac{\text{High}_t - \text{Close}_{t-1}}{\text{High}_t - \text{Low}_t}$$

$$\text{Commodity Channel Index (CCI)} = \frac{\text{Typical Price} - 20 \text{ period SMA of TP}}{.015 * \text{Mean Deviation}}$$

Typical Price (TP) = (High + Low + Close)/3, DIFF<sub>t</sub> = EMA (12)<sub>t</sub> - EMA(26)<sub>t</sub>, UP<sub>t</sub> means upward price changes while DW<sub>t</sub> is the downward price changes at time t, HighestHigh = highest high for the n period, LowestLow = lowest low for the n period.



## **2.2 Machine Learning:**

Many techniques have been developed over the years to predict stock prices and trend movement. At first, the classical regression methods were used for prediction, but since stock data is a non-stationary time series data, nonlinear machine learning techniques have also been used. Two of the most widely used machine learning techniques for stock market prediction is the Support Vector Machine (SVM) and the Artificial Neural Networks (ANN) where each algorithm has its strength and weakness in learning patterns.

### ***2.2.1 Support Vector Machine (SVM)***

The Support Vector Machine was developed in the late 1970 by Vapnik and his colleagues based on the statistical learning theory. It became a very hot topic due to its successful application in regression and classification tasks, as well as time series prediction and financial related applications [5]. Due to its chaotic, noisy and non-stationary inherent nature, the financial time series data appears to be a good candidate for a non-traditional prediction model such as the SVM.

The Support Vector Machine (SVM), has been widely used for many machine learning tasks like object classification, pattern recognition, regression analysis and time series prediction. SVR short for Support Vector Regression is the process of estimating a function from observed data which will then train the SVM.

Let  $x(t)$  be a given a set of time series data, where  $t=\{0,1,2,3\dots N-1\}$  is a series of N discrete samples, and  $y(t+\Delta)$  is some future predicted values, where  $t$  is greater than or equal to N.

By using regression analysis for time series prediction, the prediction functions for linear and non-linear regression problem are defined in equations (1) and (2):

$$f(x) = (w \cdot x) + b \quad (1)$$

$$f(x) = (w \cdot \phi x) + b \quad (2)$$

When the data is not linear in its space, Kernel functions  $\phi(x)$  are used to map the

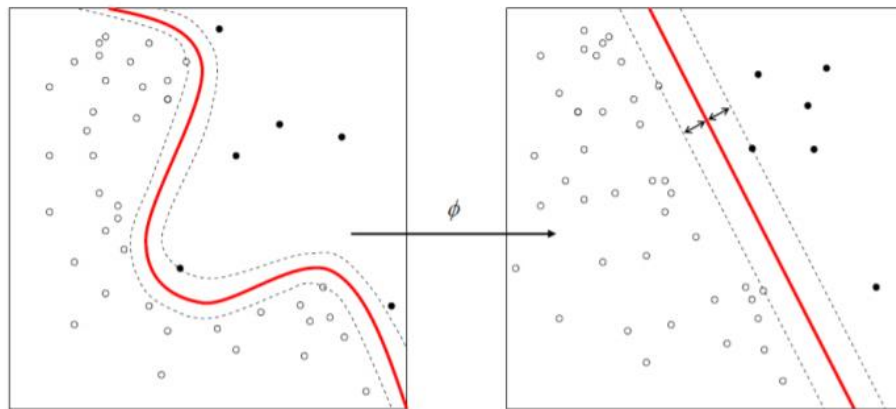


Figure 1: SVM Kernels

input data  $x(t)$  to a higher dimension space, then perform linear regression in the mapped higher dimensional space. The overall goal is to finding optimal weights  $w$  and threshold  $b$  that will minimize the regularized risk. First the weights are flattened, which can be measured by the Euclidean norm and second, minimization of the error generated by the estimation process, known as the empirical risk. [6]

### 2.2.2 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) are inspired by the functionality of the “human brain”. The basic unit of ANN is the perceptron which is a single neuron model that takes inputs values. Each input is multiplied by its corresponding weight and then combined by a non-linear function usually called activation function that generates the output. [7] The graphical structure is shown below:

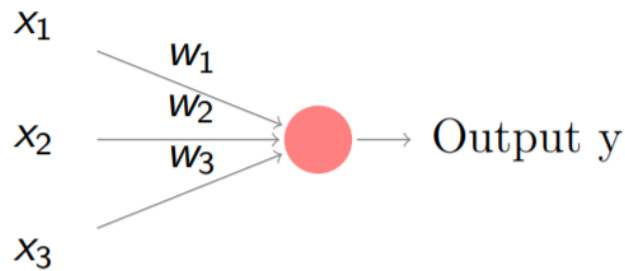


Figure 2: The Perceptron – An Artificial Neuron

The mathematical expression of a single neuron is represented by a linear combination of inputs passed to the activation function  $f$ , which is usually a sigmoid, or  $\tanh$  function:

$$y = f(\sum_{i=1}^m w_i x_i + x_0) \quad (3)$$

where  $m$  represents the number of inputs,  $w$  represents the weight and  $x_0$  is the bias. The functions for *sigmoid* and *tanh* are present below in equations 4 and 5:

$$\sigma = \frac{1}{1 + e^{-x}} \quad (4)$$

$$\tanh = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (5)$$

Multilayer Perceptron is a more sophisticated model with multiple layers, usually made of an input layer, an output layer and many hidden layers. Its graphical structure is shown in the figure below. It is a type of network, whose nodes in the same layer do not connect to each other. There is also no loop in the entire model. In addition, it can be viewed as a feed-forward network [8].

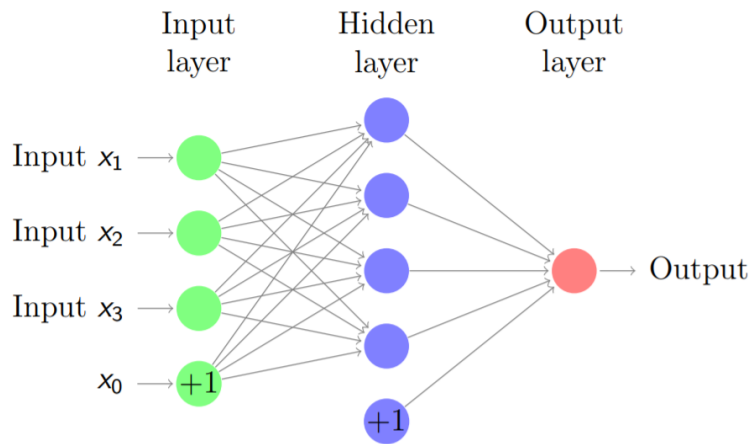


Figure 3: Typical Artificial Neural Network – Multi-layer Perceptron

When the feed-forward phase is executed to compute all the activations and the output of each layer. The difference between the output and the target values are calculated

as the cost error. Next the back-propagation phase is fired. In this process, the error will be back propagated to each layer to adjust the weights. [7] Then feed-forward and back-propagation will be further iterated until the error converges to a certain desired level.

“ANN has been successfully used for modeling and predicting financial time series” [3].

The ability of ANNs to discover non-linear relationships between data makes them ideal for the problem of stock market prediction.

### 2.2.3 Long Short Term Memory (LSTM)

Recurrent Neural network (RNN) has been receiving great attention for modeling time series and sequence problems due to internal cyclic connections between units that can store information while processing new inputs. Figure 4 shows how a RNN forms a Deep Neural Network (DNN) where information is passed from unit to another at each time step.

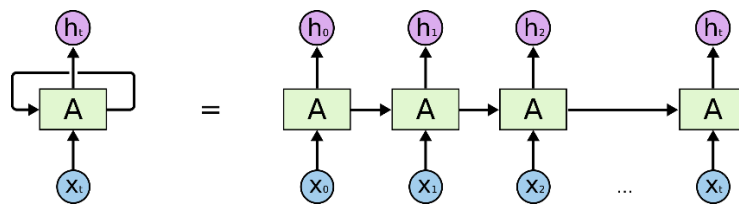


Figure 4: Recurrent Neural Network

The chained-like nature of RNN is closely related to sequences and lists. In fact, recently, RNN has been successfully applied to speech recognition, language modeling, translation and many other problems. However, RNN suffers from the Vanishing Gradient Problem, since RNN are trained by back-propagation through time and when the gradient is passed back through many time steps, it tends to grow or vanishes. This problem makes it difficult for RNN models to learn long range dependencies. [9]

Long Short-Term Memory (LSTM) is a special variant of RNN that was introduced by Hochreiter & Schmidhuber (1997) [10] and were explicitly designed to solve the vanishing gradient problem that RNN would suffer when dealing with long term data sequences. The issue is solved by incorporating a Memory Cell with three main gates: a forget gate, an input gate, and an output gate (Figure 5). The role of these gates is to update the memory in a very precise way by controlling which pieces of information to forget, what to update and what information to pay attention to.

The “forget gate” decides what information we are going to keep from the previous state that represents the long term memory and what to discard. This is determined by a sigmoid activation – formula (6) – on the previous hidden state  $h_{t-1}$  and input  $x_t$ , the output is a number between 0 and 1 where 0 means discard and 1 means keep.

$$f_t = \sigma(x_t U_f + h_{t-1} W_f + b_f) \quad (6)$$

Next we need to decide what new information we are going to keep in the cell state. This is done in two steps. First, the “input gate” decides what information we are going to

update in the cell state through a sigmoid function – formula (7). Second, we create candidate values  $\tilde{C}_t$  to be added to the state using *tanh* – formula (8).

$$i_t = \sigma(x_t U_i + h_{t-1} W_i + b_i) \quad (7)$$

$$\tilde{C}_t = \tanh(x_t U_g + h_{t-1} W_g + b_c) \quad (8)$$

After that we update the cell state by multiplying the old cell state  $C_{t-1}$  with  $f_t$  to discard what was decided to discard or forget, then we add  $i_t * \tilde{C}_t$ , which represent the new information we decided to keep by the candidate values – formula (9)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

Finally, we decide how much of the cell state we are going to output. This is the role of the “output gate” which runs a sigmoid function on the cell state and in the input – formula (10) – then applies a *tanh* function on the cell state to produce values between -1 and 1. We then multiply it with the output of the output gate to produce a filtered output of the cell state – formula (11)

$$o_t = \sigma(x_t U_o + h_{t-1} W_o + b_o) \quad (10)$$

$$h_t = \tanh(C_t) * o_t \quad (11)$$

The formulas (12) – (14) mentioned above represent a vanilla LSTM network. Other versions of LSTM exist. The “Peephole Connections” is perhaps the most popular one

introduced by Gers & Schmidhuber (2000) [11] which allows the gate layers to look to the cell state. Consequently the updated formulas for the input, output and forget gates look like:

$$f_t = \sigma(x_t U_f + h_{t-1} W_f + C_{t-1} V_f + b_f) \quad (12)$$

$$i_t = \sigma(x_t U_i + h_{t-1} W_i + C_{t-1} V_i + b_i) \quad (13)$$

$$o_t = \sigma(x_t U_o + h_{t-1} W_o + C_t V_o + b_o) \quad (14)$$

With this architecture, not only an LSTM network is able to remember information for much longer periods. It is also able to remember what's important and how long it will remain important which will enhance the performance of the model.

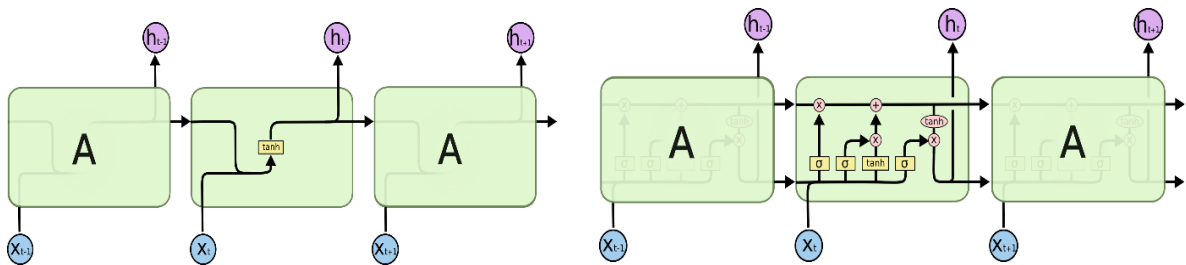


Figure 5: RNN Architecture (left) vs LSTM Architecture (right)

Since their advent, these networks have been widely used and they have shown the best results in Neutral Language processing (NLP) and in hand writing recognition where it is considered the state-of-the-art. [12]



### 2.3 State of the art

There are vast numbers of papers and articles about the subject of stock market prediction using Machine Learning, such as [3], which predicts the stock price movement of Istanbul Stock Exchange Index (ISE). The study makes a comparison between two models, one is based on artificial neural networks (ANN) and the other one is based on support vector machines (SVM). The study also involves computing ten technical indicators from stock data (open, high, low, close and volume) to be used as input features for these prediction models.

On the other hand, [1] addresses the problem of predicting stock price movement for the Indian stock markets. The study makes a comparison between four prediction models, Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive-Bayes and Random Forest using two approaches. The first approach uses ten technical indicators as continuous input values while the second approach transforms those ten technical indicators to trend deterministic data and uses them as inputs.

[4] Focuses on predicting stock market values using two Indian stock market indices namely S&P Bombay Stock Exchange (BSE) Sensex and CNX Nifty. The predictions are made for 1-10, 15 and 30 days in the future. The study proposes a two stage hybrid approach for predicting the closing price of  $(t + n)$  day from  $t^{th}$  day information data. In the first stage a Support Vector Regression (SVR) model is used to predict the value of the technical parameters for  $(t + n)$  day using the  $t^{th}$  day information. These parameters are then used in the second stage by another model (ANN, SVR or Random Forest) to predict the Closing price for  $(t + n)$  day. This hybrid approach is compared with

the general single stage approach that takes data describing the  $t^{th}$  day and directly predicts the  $(t + n)$  closing price.

[13] used LSTM networks to predict future trends of stock prices based on the price history alongside with technical analysis indicators. The experiments were conducted on different stocks from the Brazilian Stock Exchange. The results were compared with other machine learning model (Multi-Layer Perceptron and Random Forest model), and LSTM showed considerable gain in terms of accuracy.

[14] compares between different neural network models and evaluates their effectiveness on the problem of stock market prediction, these models are Multi-layer Perceptron (MLP), “Dynamic Artificial Neural Network” (DAN2) and hybrid models such as “Generalized Autoregressive Conditional heteroscedasticity” (GARCH) and Exponential GARCH (EGARCH). The comparison was done on NASDAQ index dataset, the results showed that classical MLP outperforms DAN2 and GARCH-MLP with a small difference using Mean Square Error (MSE) and Mean Absolute Deviate (MAD) measure.

## **CHAPTER 3: METHODOLOGY AND VALIDATION**

This chapter describes the approach taken in this research project for predicting stock prices. Section 3.1 describes the data used for the experiments. Section 3.2 describes the preprocessing phase of the data. Section 3.3 describes the experiments conducted comparing different prediction models with a baseline model, the results are also shown and discussed.

### **3.1 Data Collection**

Perhaps choosing data for building a stock market prediction model is not an easy task. Some papers use the historical data of a stock (Open, High, Low, Closing price and Volume). Others use technical indicators like Simple Moving Average, Weighted Moving Average, Momentum, Relative Strength Index, or fundamental indicators such as current ratio, gearing ratio, total assets, capital, long term debt, profit and loss statement such sales, depreciation, interest paid, etc. For this study, we are going to use the historical data of a stock and we are mainly going to use the Closing price.

The stocks used in this study are all retrieved from yahoo finance website using Python. The daily transactions for the last ten years from 01/10/2007 till 30/09/2017 are retrieved. As for the stocks used in this project, we randomly choose eight different stocks listed on the NYSE or NASDAQ stock market exchange. Four of these stocks: Apple Inc. (AAPL), Microsoft Corporation (MSFT), Amazon.com, Inc. (AMZN) and Alphabet Inc. (GOOGL) are extracted from the S&P 500 American stock market index made up of the largest 500 American companies. We have also selected S&P Bombay Stock Exchange

Sensitive Index (^BSESN) which is used as comparison baseline for our work, Qatar National Bank (QNBK.QA) and Ooredoo (ORDS.QA), are two local Qatari companies in addition to Qatar Exchange Index (DSM) data. The figures below shows the historical closing prices for these stocks.

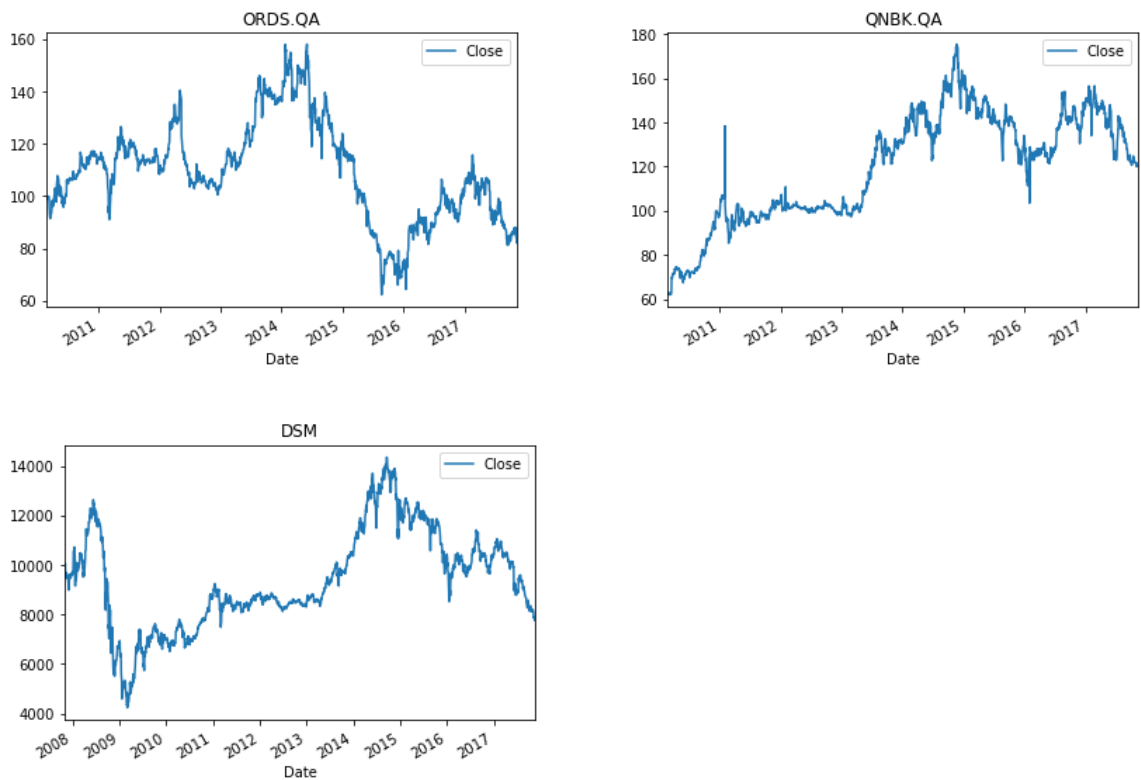


Figure 6: Historical Closing Prices for Ooredoo (ORDS.QA), Qatar National Bank (QNBK.QA) and Qatar Exchange Index (DSM)

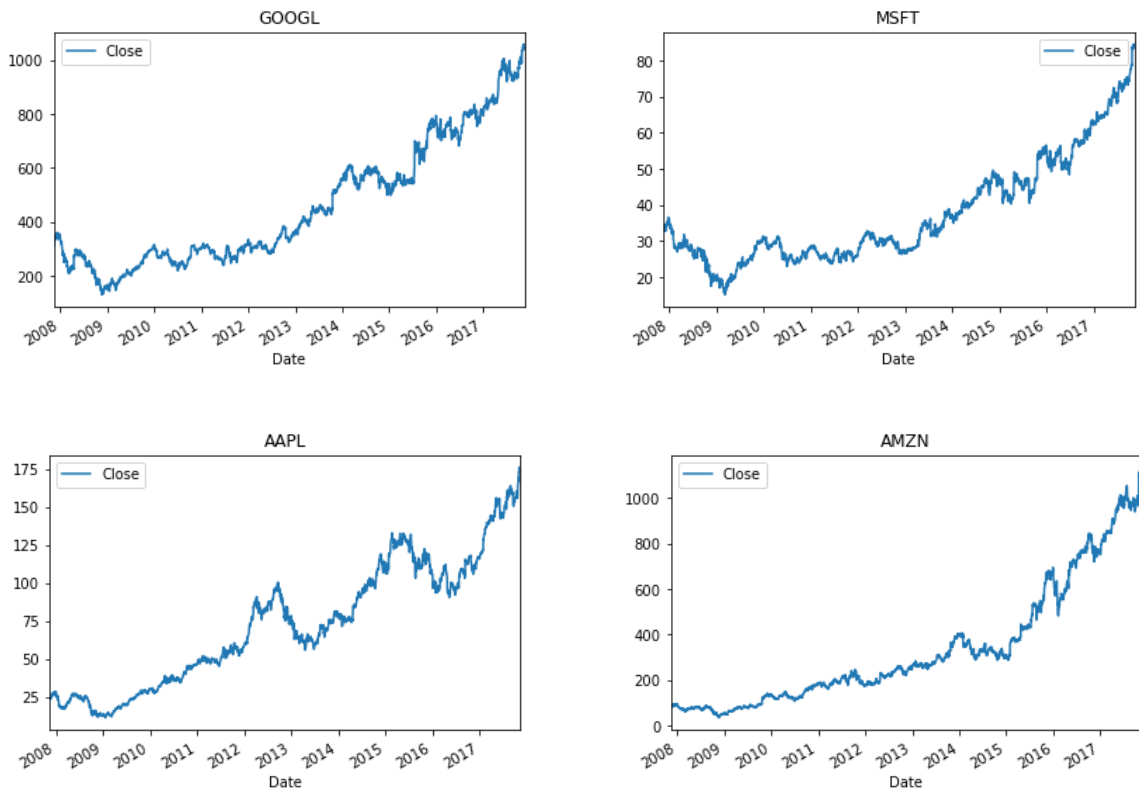


Figure 7: Historical Closing Prices for Alphabet Inc. (GOOGL), Microsoft Corporation (MSFT), Apple Inc. (AAPL) and Amazon.com, Inc. (AMZN)

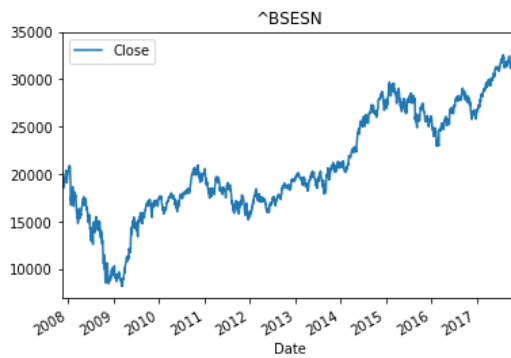


Figure 8: S&P Bombay Stock Exchange Sensitive Index (^BSESN)

## **3.2 Preprocessing**

This section describes the data preprocessing and feature extraction step in preparation for training and building the prediction models.

### ***3.2.1 Lag and Sequences Time Series Format***

Before using the data in any prediction task, the data has to be preprocessed and transformed from its raw format into the desired format. This task is a very important for any machine learning task. In this step we take the data, mainly the closing prices of the stocks and transform it from its raw time sequence format into lag and sequence (seq) format [15]. The lag represents how many time steps we want to look back in the past and this will become the features for the prediction models. As for the sequence (seq), it represents how many time steps we want to look forward in the future for forecasting and this data will become the target for the prediction models. Figure 9 and 10 shows an illustration of this transformation step. [16]

	Open	High	Low	Close	Volume
10/5/2017	154.17999	155.44	154.05	155.39	21283800
10/6/2017	154.97	155.49	154.56	155.3	17407600
...	...	...	...	...	...
10/13/2017	156.73	157.28	156.41	156.99	16394200
10/16/2017	157.89999	160	157.65	159.88	24121500
10/17/2017	159.78	160.87	159.23	160.47	18997300
10/18/2017	160.42	160.71	159.6	159.76	16374200
10/19/2017	156.75	157.08	155.02	155.98	42584200
10/20/2017	156.61	157.75	155.96	156.25	23974100
...	...	...	...	...	...
10/30/2017	163.89	168.07	163.72	166.72	44700800
10/31/2017	167.89999	169.65	166.94	169.04	36046800
11/1/2017	169.87	169.94	165.61	166.89	33637800
11/2/2017	166.60001	168.5	165.28	168.11	41393400
11/3/2017	174	174.26	171.12	172.5	59398600
...	...	...	...	...	...
11/10/2017	175.11	175.38	174.27	174.67	25145500
11/13/2017	173.5	174.5	173.4	173.97	16859000
11/14/2017	173.03999	173.48	172.12	173.07	10529348

Figure 9: Data before Transformation – AAPL Dataset

	Features							Target						
	(t-9)	(t-8)	...	(t-3)	(t-2)	(t-1)	(t)	(t+1)	(t+2)	(t+3)	...	(t+8)	(t+9)	(t+10)
10/18/2017	155.39	155.3	...	156.99	159.88	160.47	159.76	155.98	156.25	156.17	...	166.72	169.04	166.89
10/19/2017	155.3	155.84	...	159.88	160.47	159.76	155.98	156.25	156.17	157.1	...	169.04	166.89	168.11
10/20/2017	155.84	155.9	...	160.47	159.76	155.98	156.25	156.17	157.1	156.41	...	166.89	168.11	172.5
10/23/2017	155.9	156.55	...	159.76	155.98	156.25	156.17	157.1	156.41	157.41	...	168.11	172.5	174.25
10/24/2017	156.55	156	...	155.98	156.25	156.17	157.1	156.41	157.41	163.05	...	172.5	174.25	174.81
10/25/2017	156	156.99	...	156.25	156.17	157.1	156.41	157.41	163.05	166.72	...	174.25	174.81	176.24
10/26/2017	156.99	159.88	...	156.17	157.1	156.41	157.41	163.05	166.72	169.04	...	174.81	176.24	175.88
10/27/2017	159.88	160.47	...	157.1	156.41	157.41	163.05	166.72	169.04	166.89	...	176.24	175.88	174.67
10/30/2017	160.47	159.76	...	156.41	157.41	163.05	166.72	169.04	166.89	168.11	...	175.88	174.67	173.97
10/31/2017	159.76	155.98	...	157.41	163.05	166.72	169.04	166.89	168.11	172.5	...	174.67	173.97	173.07

Figure 10: Lag and Seq transformation for Closing prices - AAPL Dataset

### 3.2.2 Data Normalization

After transforming the data to the desired lead and lag format, we have to normalize (or sometimes known rescaling) the data in order to guarantee stable and fast convergence of weights and biases in the prediction models. This is a very important step in the machine

learning process and it reduces the training time by far. The simplest method is known as the Min-Max scaling, which rescales the data to a range of [0, 1] or [-1, 1]. The equation for min-max scaling is shown in (15).

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (15)$$

$$X_{changed} = \frac{X - \mu}{\sigma} \quad (16)$$

The other common approach used is the Standardization, which rescales the data to have a mean ( $\mu$ ) of 0 and standard deviation ( $\sigma$ ) of 1, as shown in equation (16).

### 3.3 Experiments

This section describes the different experiments undertaken to learn and build the prediction models Support Vector Machine (SVM), Artificial Neural Network (ANN), and Random Forest (RF), in addition to a Long Short Term Memory (LSTM) model. We also show the comparison results of evaluating these models with a baseline model developed by *Patel et al., 2015*.

#### 3.3.1 Overview

We implemented different machine learning algorithms for predicting stock market prices using python *sklearn*<sup>1</sup> library, mainly we used the Support Vector Machine (SVM) with different kernel functions since the stock market data is non-linear data, as well as Artificial Neural Network using Multi-Layer Perceptron (MLP) and Random Forest (RF).

---

<sup>1</sup> <http://scikit-learn.org/>



In addition to that, we implement a Long Short Term Memory (LSTM) Recurrent Neural Network model using *Keras*<sup>2</sup> API on top of *Tensorflow*<sup>3</sup> infrastructure.

Keras is an easy to use open source high level API for deep learning that runs on top of Tensorflow infrastructure which provides the heavy lifting. Tensorflow is an open source low level library for matrix computation and machine learning developed by Google and can run on single device or multiple devices on CPUs or GPU. [17]

The implementation process consists of multiple steps which are Data Fetch, Data Preprocessing, Training Models and Model Evaluation. Figure 11 shows an illustration of the implementation process. The process starts by downloading historical prices data for the different stocks using python APIs, the data includes date, opening price, high price, low price, closing price and volume, what we are interested in is just the date and the closing price.

---

<sup>2</sup> <https://keras.io/>

<sup>3</sup> <https://www.tensorflow.org/>

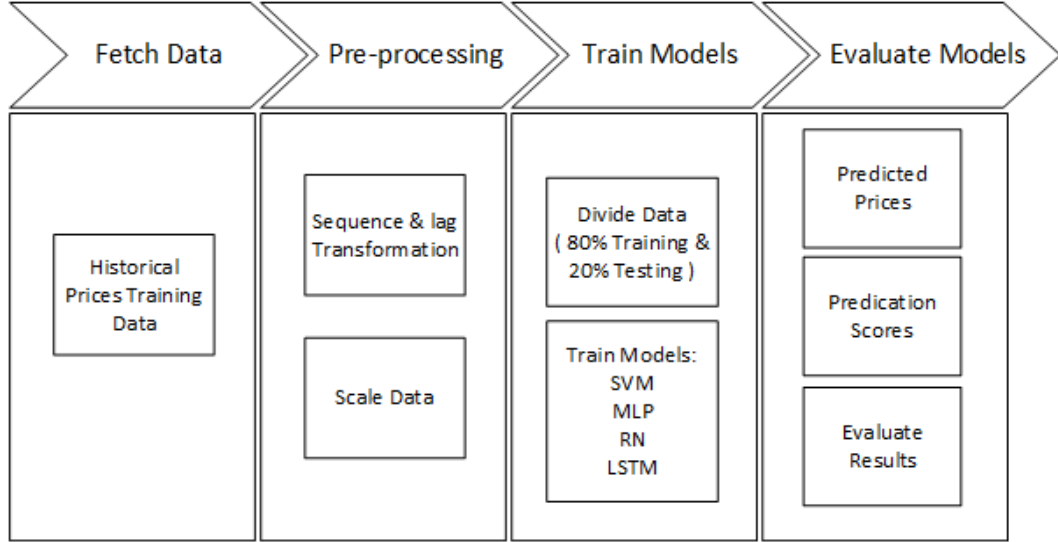


Figure 11: Methodology Overview

Data is then preprocessed and prepared for training the models. First, we remove any missing values, then we transform the data from its sequential format into lag and sequence format. Figures 9 and 10 illustrate this transformation step. After that, we rescale the data to guarantee a stable and fast training phase. The dataset was divided into training and testing sets. 80% of the dataset was used for training and 20% for testing. Then we trained our prediction models, using the lag that was generated at an earlier step as *Features* and the sequence as *Target*. The models used in the project are SVM with different kernel functions, random forest ensemble model, neural network MLP and LSTM model. Extensive experiments were conducted for each model for choosing the best configuration parameters, a summary of these models and their configuration are described in Table 1

Table 1: Models Configuration.

<b>SVM</b>				
<b>Model:</b>	<i>C</i>	<i>Gamma</i>	<i>Degree</i>	<i>Kernel</i>
<b>LinearSVR</b>	1			Linear
<b>SVR</b>	1	0.1		RBF
<b>SVR-RBF</b>	1000	4		RBF
<b>SVR-RBF-2</b>	10000	0.0001		RBF
<b>SVR-ply</b>	1000	0.1	2	Polynomial
<b>ANN</b>				
<b>Model:</b>	<i>Layers</i>	<i>Neurons per Layer</i>		<i>Solver</i>
<b>MLP-Regressor</b>	3	10, 100, 10		adam
<b>Random Forest</b>				
<b>Model:</b>	<i>Number of Estimators</i>			
<b>RN</b>	10			

The LSTM model architecture is shown in figure (12). As mentioned earlier the LSTM model is implemented using *Keras* API, the network architecture is made of two LSTM layers with a dropout layer of 20% after each LSTM layer, the dropout layer acts as regularization technique to prevent overfitting of the model by randomly dropping out units with their connections. The last layer is Dense layer which is a normal Neural network layer.

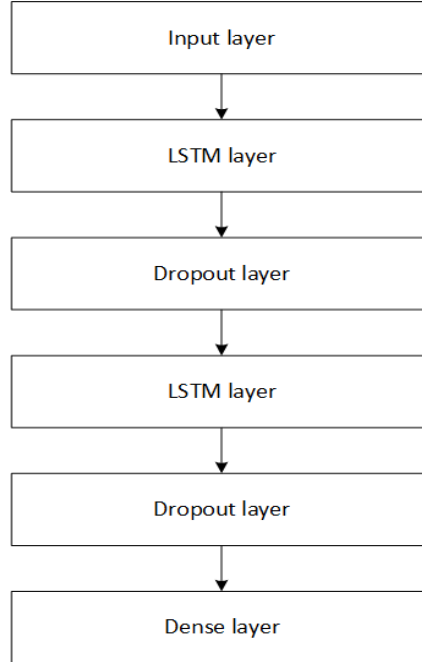


Figure 12: LSTM Model Architecture using Keras

The performance of the models were evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Square Error (RMSE) measures, the formulas of these measures are shown below:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (17)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (18)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (19)$$

where  $y_i$  is the observed value and  $\hat{y}_i$  is the forecasted value.

### 3.3.2 Experiment Baseline

In addition to the prediction models described in the previous section, we also used results of the hybrid model developed by *Patel et al., 2015* [4] as a baseline for comparison of our experimental study. *Patel et al., 2015* developed a two stage hybrid model for predicting future prices of a stock using technical indicators and historical prices, in the first stage an SVR model is used to predict the technical indicator value for  $(t + n)$  day from  $t^{th}$  day information data. In the second stage, they used the information for the first stage as input to another prediction model mainly they used an ANN, SVR and a Random Forest model to predict  $(t + n)$  closing price of a stock. We used the results reported in their research for their best prediction model which is the SVR-ANN hybrid model. Figure 13 illustrates the two stage hybrid prediction model of *Patel et al., 2015*.

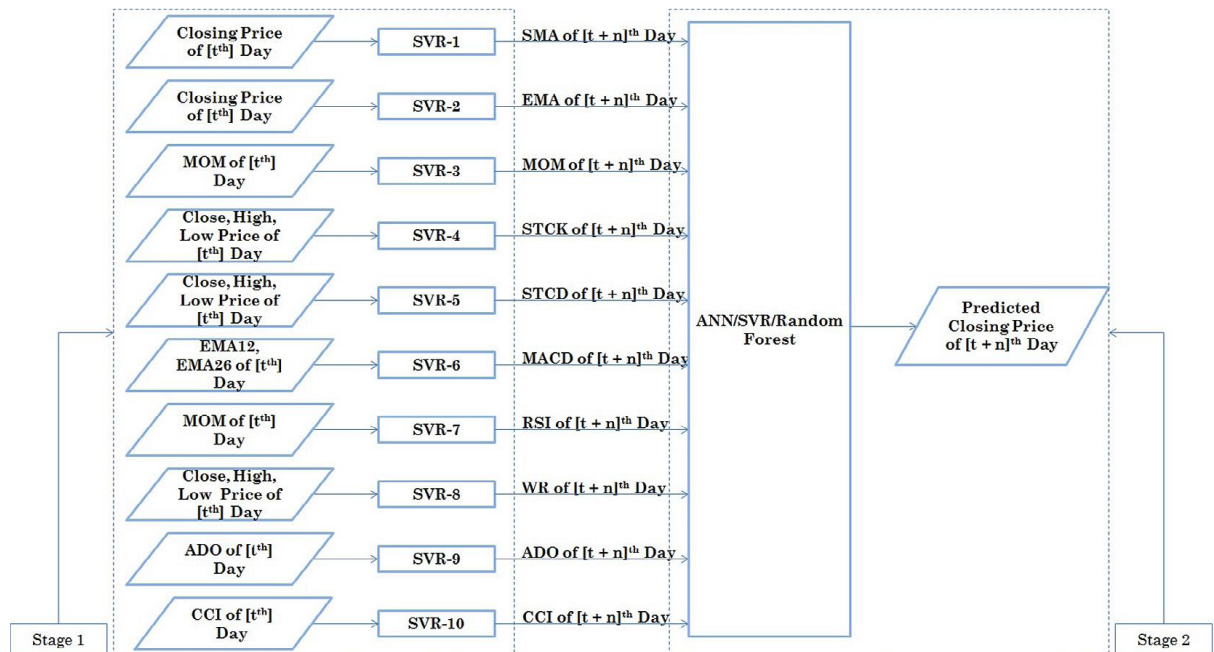


Figure 13: Two Stage Hybrid Prediction Model of Patel et al., 2015 [4]

### 3.3.3 Results

In order to have an accurate and reliable comparison results, we used the same dataset as the baseline, this dataset is made of 10 years of historical data for *S&P Bombay Stock Exchange Sensitive Index (^BSESN)* from January 2003 till December 2012. Tables 2, 3 and 4 show the results of the different prediction models and the baseline for predicting ten days in the future using MAE, MSE and RMSE measures. The values in bold represent the best value of a given model for the forecasted day.

Table 2: MAE Results for Prediction Models and the Baseline

Model	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10
LinearSVR	<b>155.44</b>	231.97	291.51	338.45	381.19	411.83	446.17	469.76	494.96	522.26
MLP-Regressor	340.69	368.98	429.68	500.04	432.34	493.98	500.06	530.19	553.20	557.27
RN	183.19	249.44	299.70	343.55	379.34	407.47	436.31	460.06	488.49	517.69
SVR	231.45	282.81	326.86	361.34	394.30	423.83	447.46	470.78	489.29	515.74
SVR-ply	6088.60	6017.90	6191.42	6099.54	6161.29	6105.61	6274.42	6129.71	6137.45	6103.07
SVR-RBF	722.02	732.46	746.74	778.42	784.01	811.46	819.96	853.53	906.64	965.39
SVR-RBF-2	160.44	232.10	293.91	336.59	377.38	411.51	445.76	465.40	494.67	524.37
LSTM	155.74	<b>228.84</b>	<b>288.35</b>	<b>333.50</b>	<b>373.49</b>	<b>407.31</b>	<b>435.33</b>	<b>458.48</b>	<b>485.80</b>	512.25
Baseline	272.71	296.41	354.01	392.53	403.21	429.90	445.19	459.58	499.37	<b>474.62</b>

Table 3: MSE Results for Prediction Models and the Baseline

Model	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10
<b>LinearSVR</b>	<b>39159.62</b>	84411.68	134671.8	177957.8	224428.6	264922.8	313040.4	354638.7	395046.9	438375.3
			5	3	8	9	4	9	2	0
<b>MLP-Regressor</b>	179122.8	207270.5	281183.3	408394.6	291977.2	391043.3	399113.2	443707.4	489932.7	494638.7
	1	2	0	9	7	4	0	0	2	7
<b>RN</b>	57362.60	98189.84	140454.5	182258.3	221364.1	<b>258210.8</b>	<b>296448.0</b>	<b>338591.1</b>	<b>384991.5</b>	434434.0
			5	7	2	<b>5</b>	<b>8</b>	<b>5</b>	<b>4</b>	5
<b>SVR</b>	91130.00	127688.2	165959.1	199840.1	237080.2	275924.5	310750.3	350714.6	389083.1	432388.1
		4	1	3	4	2	9	8	0	5
<b>SVR-ply</b>	7396672	7127224	7630959	7351782	7558595	7389175	7888835	7451927	7444787	7364326
	7.17	3.21	0.40	2.16	9.91	0.48	4.43	4.69	3.71	9.72
<b>SVR-RBF</b>	915608.6	927516.0	952088.1	1025647.	1025521.	1072575.	1073859.	1149588.	1264226.	1424692.
	3	7	9	20	38	08	99	53	05	39
<b>SVR-RBF-2</b>	42217.48	84410.03	135100.6	176363.0	219514.3	265138.2	312315.0	348154.7	397017.1	442273.6
			5	4	6	9	1	0	8	4
<b>LSTM</b>	39231.75	<b>82501.03</b>	<b>131735.0</b>	<b>173590.6</b>	<b>216521.2</b>	260802.2	303138.4	342904.0	385581.7	427388.9
			9	7	6	8	5	5	8	0
<b>Baseline</b>	118395.0	146339.7	212027.6	258336.4	259749.8	312341.9	320178.0	367846.1	411681.5	<b>387086.1</b>
	9	0	2	2	8	9	9	8	7	<b>3</b>

Table 4: RMSE Results for Prediction Models and the Baseline

Model	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10
<b>LinearSVR</b>	<b>197.89</b>	290.54	366.98	421.85	473.74	514.71	559.50	595.52	628.53	662.10
<b>MLP-Regressor</b>	423.23	455.27	530.27	639.06	540.35	625.33	631.75	666.11	699.95	703.31
<b>RN</b>	239.50	313.35	374.77	426.92	470.49	<b>508.14</b>	<b>544.47</b>	<b>581.89</b>	<b>620.48</b>	659.12
<b>SVR</b>	301.88	357.33	407.38	447.03	486.91	525.29	557.45	592.21	623.77	657.56
<b>SVR-ply</b>	8600.39	8442.29	8735.54	8574.25	8694.02	8596.03	8881.91	8632.45	8628.32	8581.57
<b>SVR-RBF</b>	956.87	963.08	975.75	1012.74	1012.68	1035.65	1036.27	1072.19	1124.38	1193.60
<b>SVR-RBF-2</b>	205.47	290.53	367.56	419.96	468.52	514.92	558.85	590.05	630.09	665.04
<b>LSTM</b>	198.07	<b>287.23</b>	<b>362.95</b>	<b>416.64</b>	<b>465.32</b>	510.69	550.58	585.58	620.95	<b>653.75</b>
<b>Baseline</b>	8600.39	8442.29	8735.54	8574.25	8694.02	8596.03	8881.91	8632.45	8628.32	8581.57

Figures 15 – 23 show the accuracy of the prediction models of the actual closing prices compared with the forecasted prices for the next day and day ten in the future. The y-axis represents the Closing Price and the x-axis represents the Day. In Appendix B, one may find the figures for all the forecasted future closing prices (t+1, t+2 ... t+10).

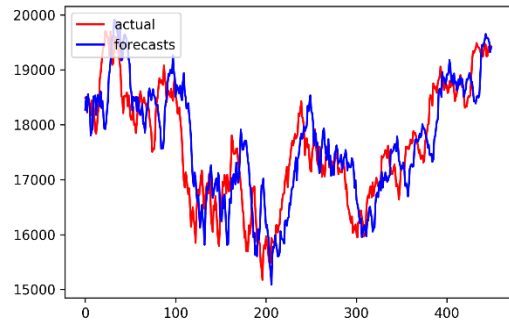
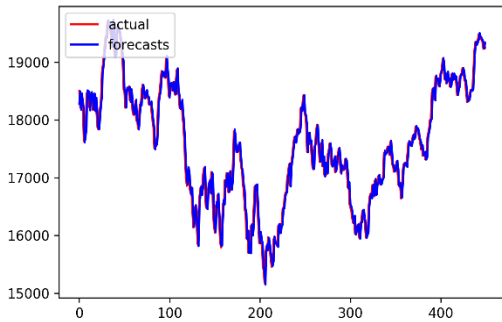


Figure 15: LSTM Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

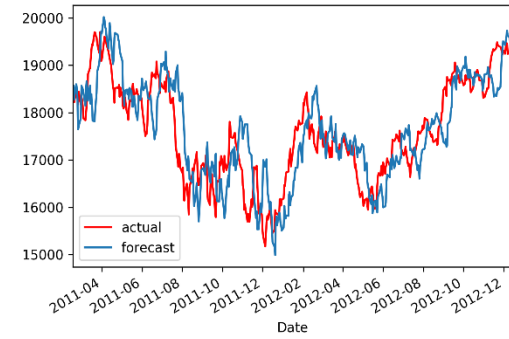
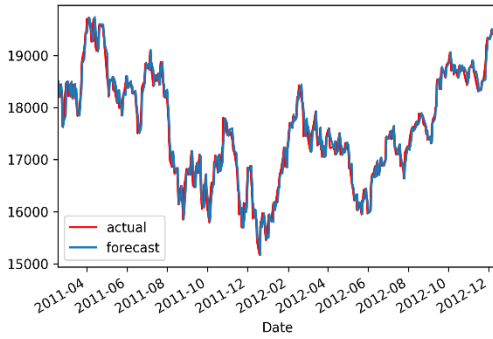


Figure 16: LinearSVR Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

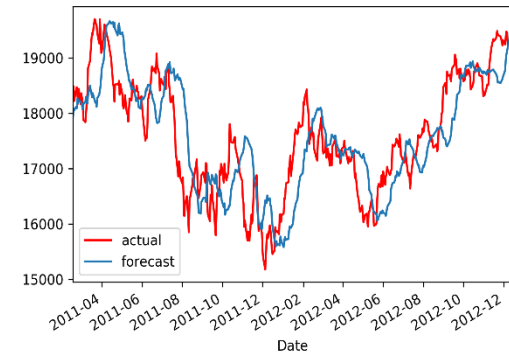
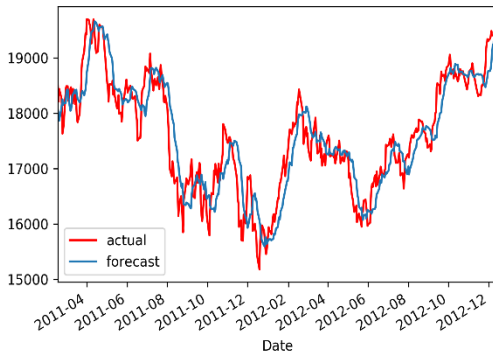


Figure 17: MLP Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days



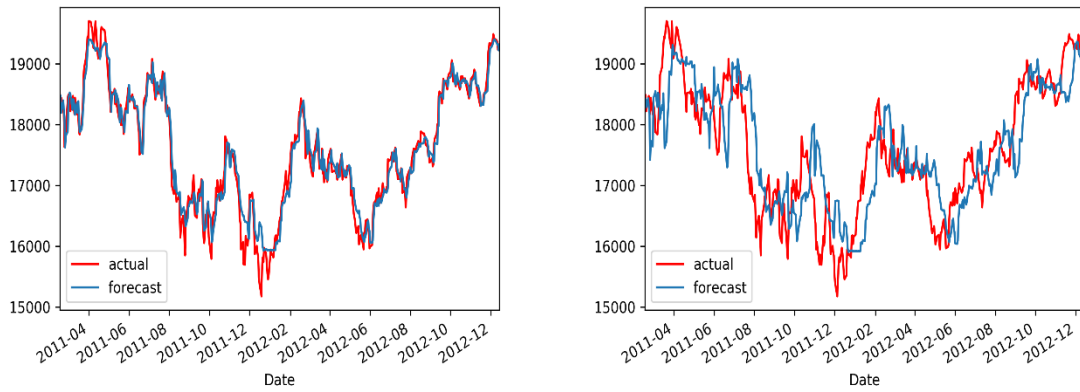


Figure 18: RN model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

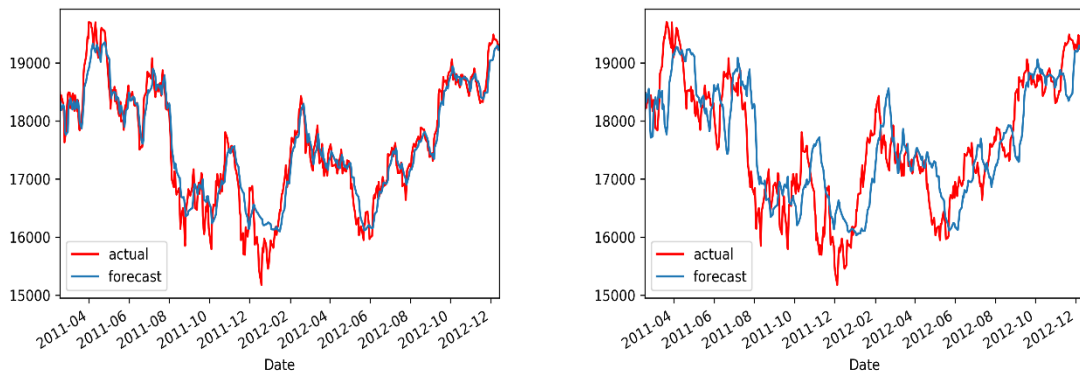


Figure 19: SVR Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

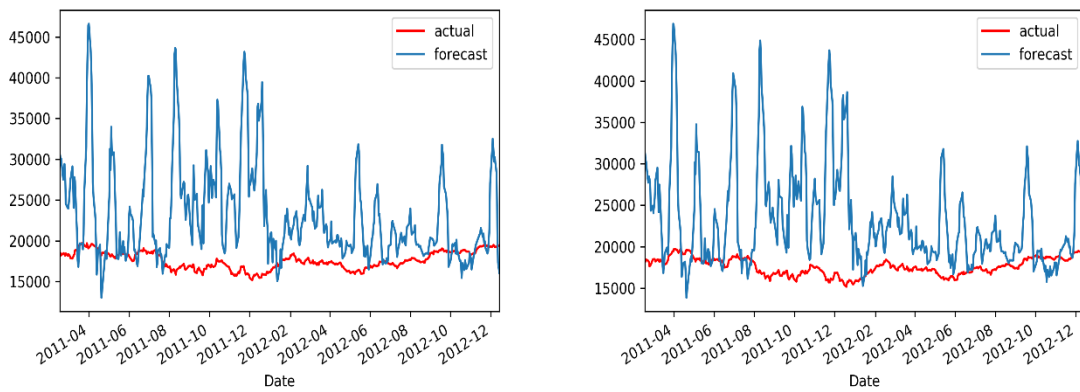


Figure 20: SVR-ply Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

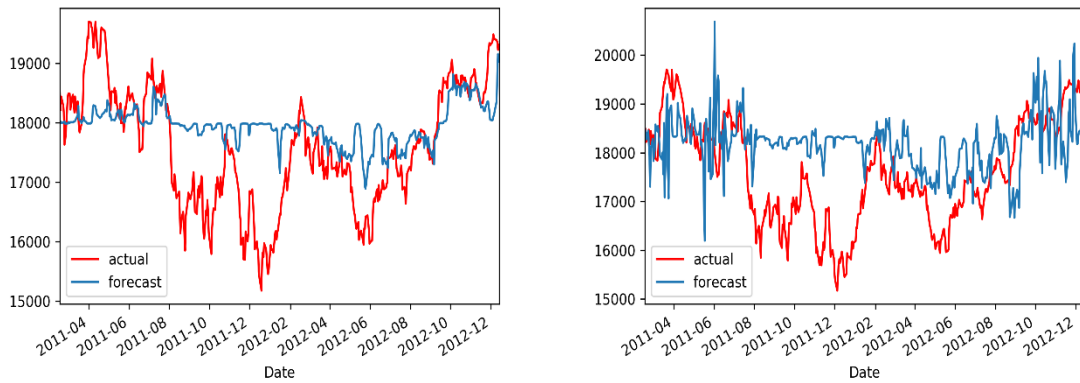


Figure 21: SVR-RBF Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

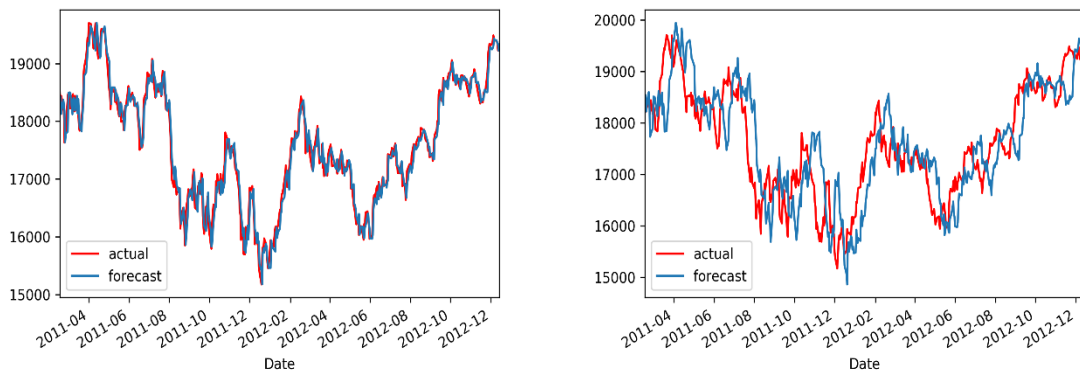


Figure 22: SVR-RBF-2 Model – Actual Closing Prices vs Forecasted for “t+1” (left) and “t+10” (right) Days

### 3.3.4 Discussion

It may be obvious for any prediction system that as prediction are made for more days in the future, error values will increase and consequently accuracy will decrease. This was evident in the results presented in the previous section.

The results have shown that the LSTM model outperforms other prediction models and the baseline using MAE measure for the ten forecasted days except for t+1 where

LinearSVR model performed better LSTM model. But it is worth mentioning that the difference between the two models is negligible (155.44 and 155.74), as for  $t+10$  results, the baseline model outperformed all the models. For MSE and RMSE measures, no single model outperformed the other for days of prediction, the results varied between models and for a prediction day. Accordingly, for  $t+1$  results LinearSVR performed better than the other models, for  $t+2$  to  $t+5$  results LSTM was better whereas Random Forest (RN) was better for results of  $t+6$  to  $t+9$ . For  $t+10$  results, LSTM was the best with MSE measure but the baseline model performed better than LSTM and the other using RMSE measure. For more details about the results, refer to tables 2-4. Figures 15 – 22 show the actual price values of *BSESN* stock compared with forecasted values using the prediction models for day one ( $t+1$ ) and day ten ( $t+10$ ), the visual representation shows the effectiveness of LSTM model in addition to LinearSVR, RN and SVR-RBF-2.

The results have shown that, although LSTM are complex and hard to configure and train, they can perform better than traditional machine learning algorithms and is more suitable for time series problems such as the stock market forecasting. Based on this, and building upon these results, we have developed a stock forecasting application with LSTM model as the core engine for prediction.

## **CHAPTER 4: CASE STUDY – DAILY STOCK PREDICTION APPLICATION**

This chapter describes the implementation of the Daily Stock Prediction (DSP) web application based on the LSTM prediction model developed in this project. Section 4.1 presents an overview of the system and different components. Section 4.2 addresses the advantages and the applicability of such system. Section 4.3 talks about the limitations in the system.

### **4.1 System Overview**

Motivated by the idea that forecasted stock prices will provide investors with more confidence in making trading decisions, we developed an application that will forecast future prices of selected stocks. The DSP application core uses an LSTM prediction model trained every day on historical stock data to predict prices for ten days in the future.

The system has two main components, a Forecasting Engine and a Web Application. The Forecasting Engine is a schedule job that will run every day. It will download the historical prices for a pre-configured list of stocks from Yahoo Finance. Downloaded data will be preprocessed and prepared to train the LSTM model. After that, the LSTM model is trained on the processed data and predict stock closing price for the next ten days. Consequently, for every day the Forecasting Engine will run and forecast tomorrow's price and the price of the next ten days prices for a given stock. These forecasted prices are stored in a database to be retrieved and displayed by the Web

Application. The Web Application is the front end of the system which displays the pre-configured list of stocks with tomorrow’s predicted price and the trend whether the price is going up or down as a quick view. Users can also select a stock from the list to view the ten days forecasted prices and a trending chart to get a glance of how the prices are changing in the next ten days. Figure 23 shows an overview of the system and the interaction of the different components.

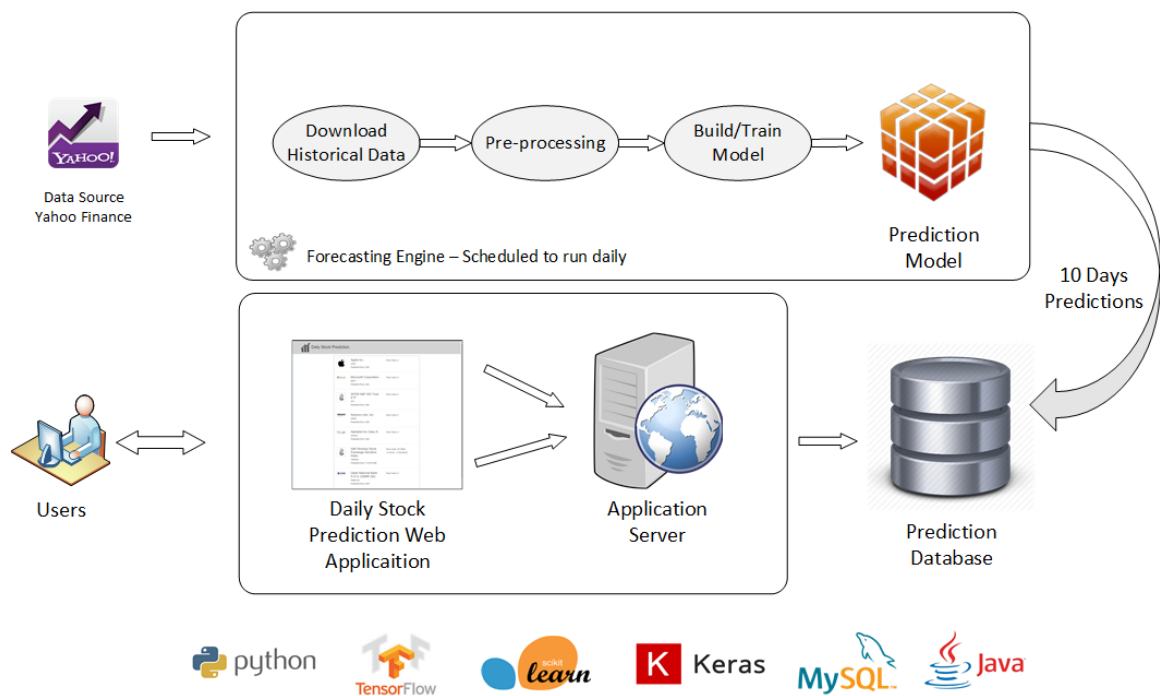


Figure 23: System Overview – Daily Stock Prediction Application

The system is implemented using different open source technologies. For the Forecasting Engine, we used python and Keras Framework for building the LSTM model

that runs on top of Tensorflow infrastructure which provides the heavy lifting of the computation. For the database we used MySQL RDMS and for the Web Application we used Java EE running on GlassFish application server.

The application is easy to use with simple design and rich content. The landing page of the application shows a pre-defined list of stocks – figure 24 – that shows the details of each stock mainly the name of the stock, ticker symbol and the logo of the company in addition to the next day's predicted closing price, on the left side of the list. The trend movements of the predicted price are displayed based on the difference between the predicted price and the previous closing price. If the trend is moving up, it is highlighted in green with an upward arrow. If the trend is moving down it is highlighted in red with downward arrow in addition to percentage difference with respect to the previous closing price. The idea behind this list is to quickly identify the trending stocks to help the users to choose from for investment if they wish to.










Daily Stock Prediction		
	<b>Apple Inc.</b> AAPL Predicted Price: 175.76 USD	<b>▲ 0.79 (0.452 %)</b> Predicted Close vs Previous Close 175.76 174.97
	<b>Microsoft Corporation</b> MSFT Predicted Price: 84.06 USD	<b>▲ 0.80 (0.961 %)</b> Predicted Close vs Previous Close 84.06 83.26
	<b>SPDR S&amp;P 500 Trust ETF</b> SPY Predicted Price: 258.10 USD	<b>▼ -2.26 (-0.868 %)</b> Predicted Close vs Previous Close 258.10 260.36
	<b>Amazon.com, Inc</b> AMZN Predicted Price: 1125.23 USD	<b>▼ -60.77 (-5.124 %)</b> Predicted Close vs Previous Close 1,125.23 1,186.00
	<b>Alphabet Inc Class A</b> GOOGL Predicted Price: 1047.34 USD	<b>▼ -9.18 (-0.869 %)</b> Predicted Close vs Previous Close 1,047.34 1,056.52
	<b>S&amp;P Bombay Stock Exchange Sensitive Index</b> *BSESN Predicted Price: 33310.83 INR	<b>▼ -368.408 (-1.094 %)</b> Predicted Close vs Previous Close 33,310.83 33,679.238
	<b>Qatar National Bank S.A.Q. (QNBK.QA)</b> QNBK.QA Predicted Price: 120.77 QAR	<b>▲ 4.77 (4.112 %)</b> Predicted Close vs Previous Close 120.77 116.00
	<b>Ooredoo Q.P.S.C. (ORDS.QA)</b> ORDS.QA Predicted Price: 84.99 QAR	<b>▲ 1.59 (1.906 %)</b> Predicted Close vs Previous Close 84.99 83.40
	<b>Qatar Exchange Index</b> DSM Predicted Price: 8155.68 QAR	<b>▲ 329.91 (4.216 %)</b> Predicted Close vs Previous Close 8,155.68 7,825.77

Figure 24: Trending Stocks List

Users can click on the logo of any stock from the list to view more details about the forecasted closing price for ten days in the future. A different page is displayed that shows the details of stock with trending description same as the list view, in addition to the

forecasted ten days prices and at the bottom of the page, a chart shows the forecast price trend movement as shown in figure 25.

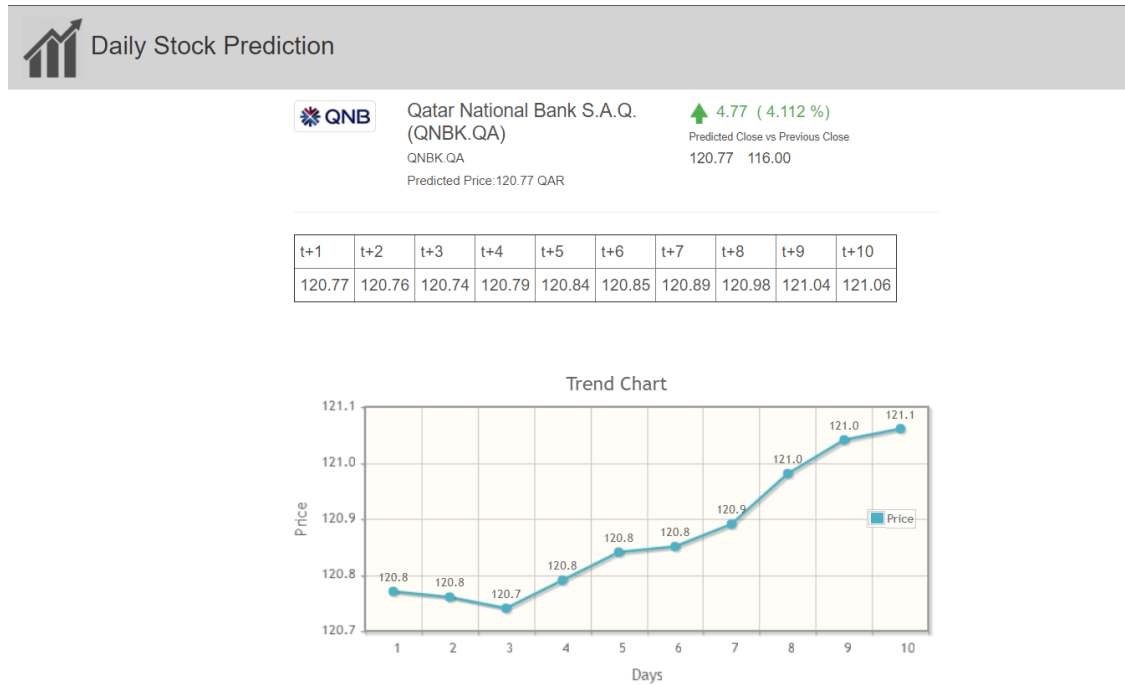


Figure 25: 10 Days Price Forecasts and Trending Direction

The motivation behind this page is to visualize the forecasted price trend movement of a stock which will give users more confidence and better trading decisions for buying or selling investments.



## **4.2 System Applicability and Usability**

In the world of trading and investment, the key factor to maximize profit and reduce risk, is to identify a trend early and accurately. For a long time, technical analysis and charting was the main approach for identifying trends, but technical analysis is complex and it takes years of experience to become successful.

Having a system - like the DSP Web Application - that is to use and can predict future stock prices with high accuracy based on learning algorithms would bring a lot of benefits to users like which stock to pick and when to invest or trade. Users will no longer need experts to help them with their investments. A system that can predict future prices and trends will help them to develop their own trading strategies. They can choose to invest in a stock when the trend is going up or they can sell when the trend is going down. This will give users a significant edge in the market and increase their chances of success. When it comes to investment in the stock market, human decisions are influenced by emotional bias to a certain stocks or sometime decisions are driven by intuitions. The system will help users to successfully pick stocks for investments by showing them top trending stocks or those stocks with the highest returns. This will not only help them to be more successful, but will remove that emotional bias form influencing their decisions.

## **4.3 System Limitations**

Despite the advantages the DSP web application can offer to users, there are some limitations in the system. These limitations are listed below:

- The predication models are re-trained everyday using the last ten years of historical data for every stock available in the database. This process is time consuming. It takes around 2.5 minutes per stock data on the current system hardware<sup>4</sup>. As the list grows, building prediction models for all the available stocks might not finish on time to be ready for next day's forecasting operations. One way to overcome this limitation is to add more computational resources to system or using distributed processing to speed up the process.
- Currently the historical stock data are retrieved from Yahoo Finance Website using open source python API, when making many API calls the service blocks and fails to retrieve the stock data and as a result we have to wait few seconds for the connection to reset to resume downloading. To overcome this limitation, a more reliable data source should be used and perhaps a paid subscription data source like Bloomberg is preferable since it also provides access to wide range to stocks.
- Currently, price predictions are made for ten future days. These predictions are good for short-term investments decisions. However, if investors are looking for long-term investments this is not enough as it does not give enough insights for investment decisions. One way to improve this limitation is to stretch the forecasting to 30 or 45 days in the future.

---

<sup>4</sup> Intel Core i7 2.9 GHz with 4 cores, 32 GB RAM, "NVIDIA Quadro M2000M" GPU with 640 cores and 4 GB RAM

## CHAPTER 5: CONCLUSIONS AND FUTURE WORK

In this last chapter, the motivations and the problems we aimed to address in the scope of this project are briefly concluded in section 5.1 in addition main contribution of this work. In section 5.2, we talk about the future work directions that might be taken to improve and build upon this work.

### 5.1 Conclusion

Stock market forecasting has attracted researchers for many years, nevertheless the problem is still considered challenging and unsolved, with every new developed algorithm in the domain of machine and or recently in deep learning, the stock market forecasting problem is put again into the test. Long-Short Term Memory (LSTM) a subsequent version of Recurrent Neural Network (RNN) has recently gained wide attention for its good performance with time series problems. In this project, we have tackled the problem of stock market forecasting using machine learning and we also put deep learning through LSTM into the test. We have developed a number of prediction models using known machine learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN) and Random Forest (RN) in addition to an LSTM model. We have also compared the performance of these models with a baseline model developed by *Patel et al., 2015* in predicting stock prices for ten days in the future. The results have shown that LSTM has outperformed other prediction models and the baseline model for the first nine days using MAE measure except for the first day where SVM was the best and for day ten the baseline model was better. Also the empirical results of the accuracy for the forecasted

values in comparison with the actual values are almost identical and most importantly the trend movement of the prices are well learned and clearly captured. However, the accuracy becomes less and less as predictions are made for more days in the future. We have taken the outcome of this research project to a further step by developing an application for stock prediction and using the LSTM prediction model as the core engine of the application for forecasting. The motivation behind developing this application is to provide users with insights about stock prices trend movement which will help them in making better investment decisions and more confidence in developing trading strategies.

## **5.2 Future Work Directions**

In the project we have explored the stock market forecasting problem using historical stock prices data only, but we know that stock prices are influenced by many external factors such as the general economic conditions, firms' policies and financial state, political events, investors' expectation, etc. One way to get a general understanding of the external factors impact on the stock movement is to analyze the daily news using Sentiment Analysis or sometimes known as Opinion Mining. Including Sentiment Analysis will add an additional dimension of features to the forecasting process and will eventually improve the accuracy. Another improvement that might be worth investigating is to include some technical indicators in the prediction model especially those technical indicators that are directly related to prices like Simple Moving Average or Weighted Moving Average.

In this project, we did not touch base on the applicability of these results on trading. The intension was to develop some trading algorithms based on the forecasted prices and

simulate those strategies using Backtesting and trading platforms such as *Quantopian*<sup>5</sup> to see the viability of these results.

The Daily Stock Prediction Application can bring a lot of benefits to users, some of the suggested future improvements to the system could be:

- Implement a mobile version of the application to reach out to wider range of users which will improve usability of the application and user engagement especially for those user who spend more time on mobile devices. This will create a direct channel with the user and make the information available at their fingertips.
- Change the data source of historical stock information from open source to more reliable sources with reliable APIs such as Bloomberg which provides a broader range of access to stocks in addition to other features such as news feeds and real-time price quotes.
- Stretch the forecasting to 30 or 45 days in the future instead of 10 days to allow users to plan for long term investments.

---

<sup>5</sup> <https://www.quantopian.com/>

## REFERENCES

- [1] J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259-268, 2015.
- [2] B. G. Malkiel, *A random walk down Wall Street: including a life-cycle guide to personal investing*, WW Norton & Company, 1999.
- [3] Y. Kara, M. A. Boyacioglu and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311-5319, 2011.
- [4] J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, p. 2162–2172, 2015.
- [5] P. D. Yoo, M. H. Kim and T. Jan, "Machine learning techniques and use of event information for stock market prediction: A survey and evaluation," in *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, 2005.
- [6] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, 2009.

- [7] A. Prastyo, D. Junaedi and M. D. Sulistiyo, "Stock price forecasting using artificial neural network:(Case Study: PT. Telkom Indonesia)," in *Information and Communication Technology (ICoIC7), 2017 5th International Conference on*, 2017.
- [8] A. M. F. Souza and F. M. Soares, *Neural network programming with Java*, Packt Publishing Ltd, 2016.
- [9] M. Långkvist, L. Karlsson and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11-24, 2014.
- [10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735-1780, 1997.
- [11] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, 2000.
- [12] A. C. M. P. R. A. d. O. David M. Q. Nelson, "Stock markets price movement prediction with LSTM neural networks," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1419-1426, 2017.
- [13] D. M. Q. Nelson, A. C. M. Pereira and R. A. d. Oliveira, "Stock Market's Price Movement Prediction With LSTM Neural Networks," *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.
- [14] E. Guresen, G. Kayakutlu and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, pp.

10389-10397, 2011.

- [15] K. Ganguly, *R Data Analysis Cookbook - Second Edition*, Packt Publishing, 217.
- [16] J. Brownlee, "Time Series Forecasting as Supervised Learning," 2016. [Online].  
Available: <https://machinelearningmastery.com/time-series-forecasting-supervised-learning/>.
- [17] A. Gulli and S. Pal, *Deep Learning with Keras*, Packt Publishing, 2017.
- [18] M. Ballings, D. V. d. Poel, N. Hespeels and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046-7056, 2015.
- [19] H. Sak, A. Senior and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.



# APPENDIX

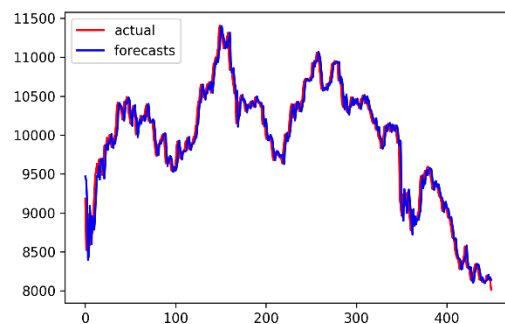
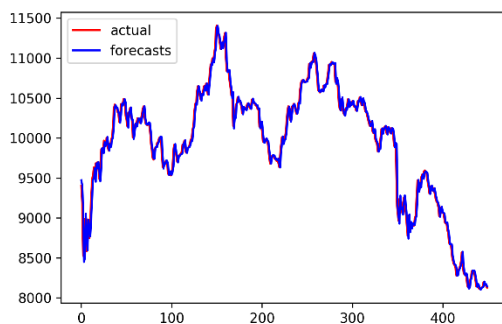
## APPENDIX A: QATAR EXCHANGE INDEX (DSM)

In this appendix we present the results of the experiments conducted on the Qatar Exchange Index (DSM) dataset using our LSTM prediction model. Table 5 shows the prediction performance evaluation results. The dataset consists of ten years of historical pricing data (2514 records) starting from 01/11/2007 till 16/11/2017.

Table 5: LSTM Evaluation Results of DSM Dataset

	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10
<b>MAE</b>	70.03	106.71	136.04	163.29	186.38	206.19	225.22	244.90	263.72	280.68
<b>MSE</b>	11020.21	23898.19	35971.85	47204.38	60236.50	73190.57	85839.87	99398.38	115099.14	128180.03
<b>RMSE</b>	104.98	154.59	189.66	217.27	245.43	270.54	292.98	315.28	339.26	358.02

Figures below show the actual closing price values compared with the forecasted values for ten days forecasts.



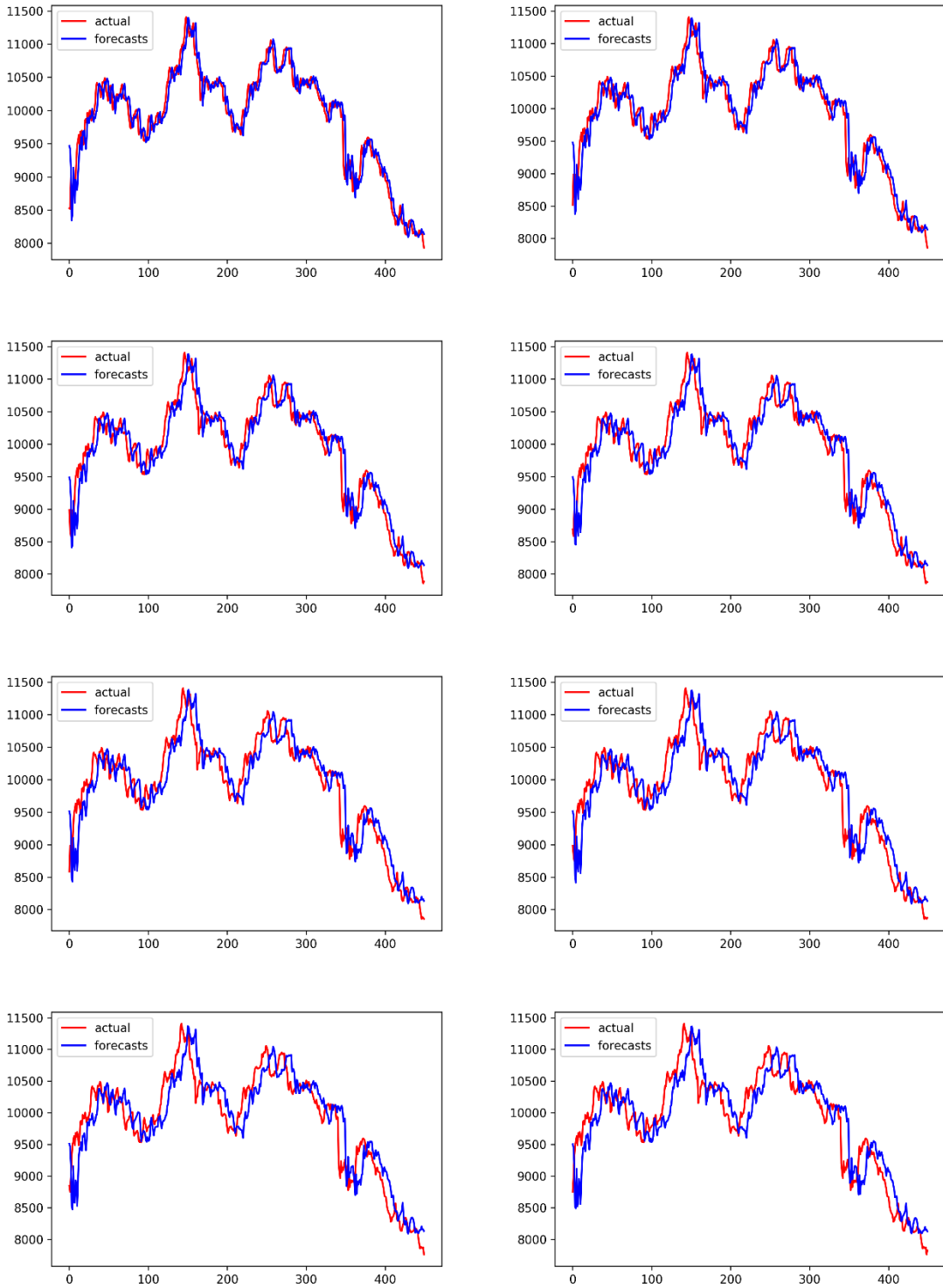
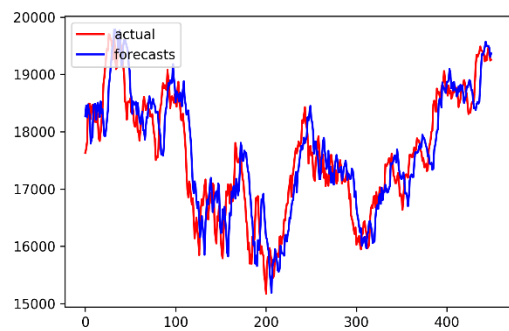
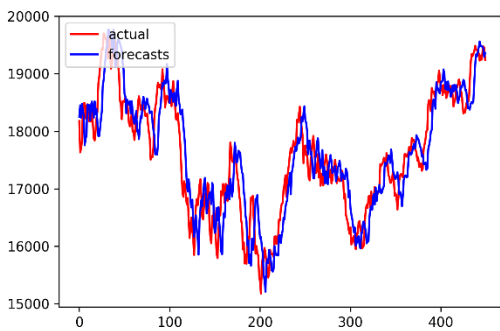
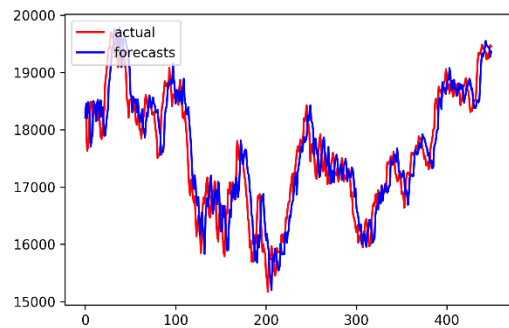
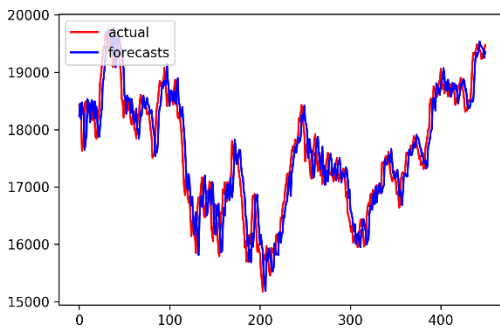
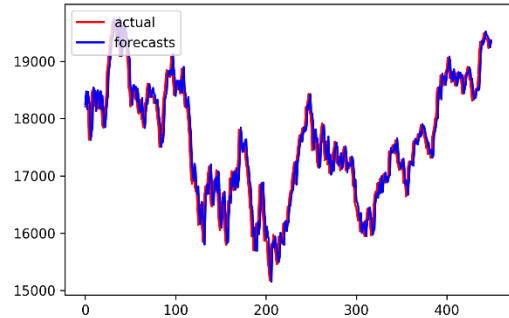
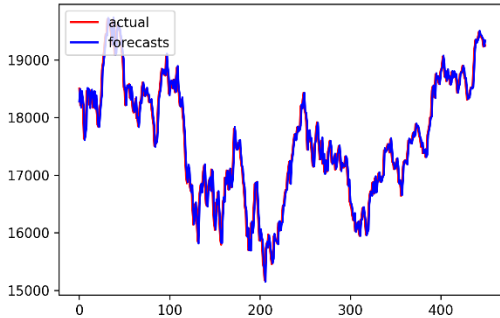


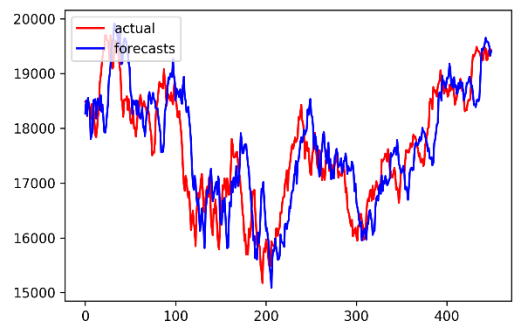
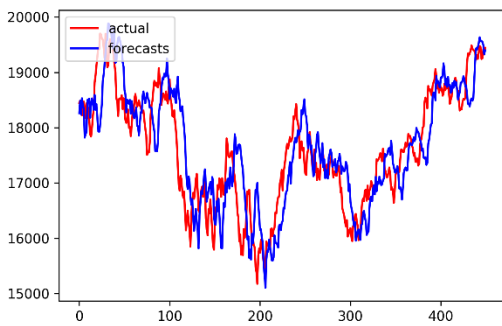
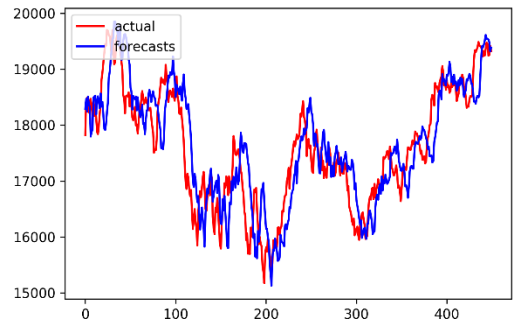
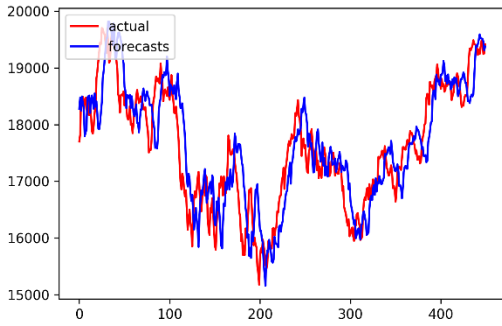
Figure 26: DSM Actual Prices vs Forecasted Prices for 10 days forecasts

## APPENDIX B: Bombay Stock Exchange Sensitive Index (BSESN)

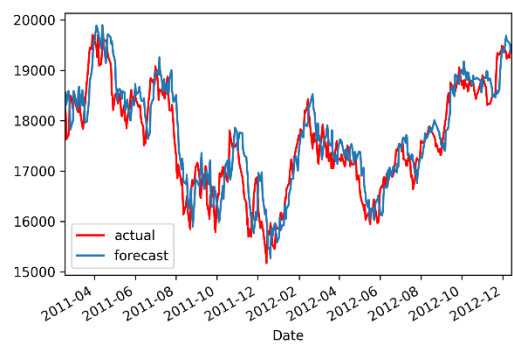
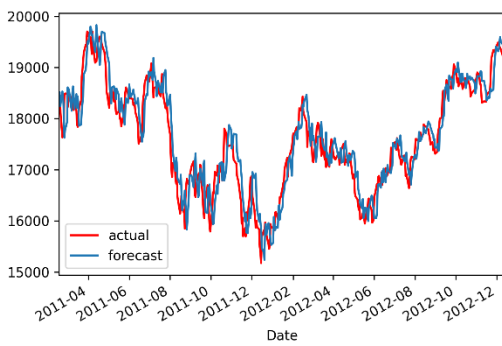
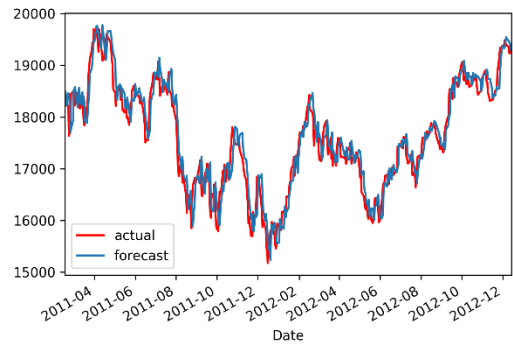
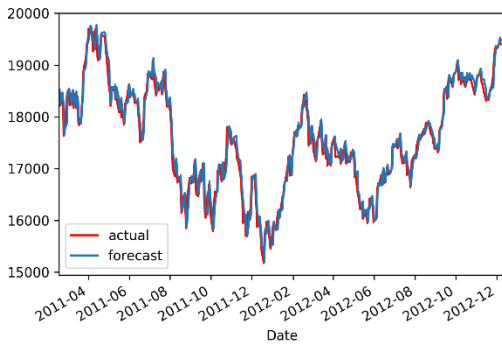
Complete results of all the prediction model for actual prices for *BSESN* stock compared with the forecasted prices for ten days forecasts.

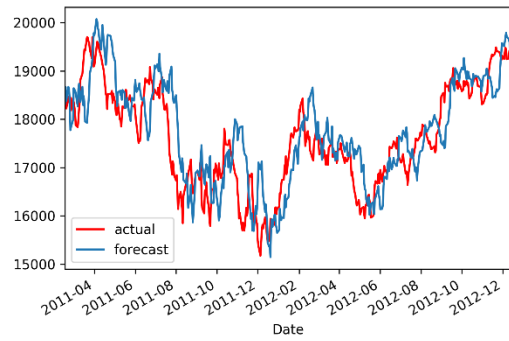
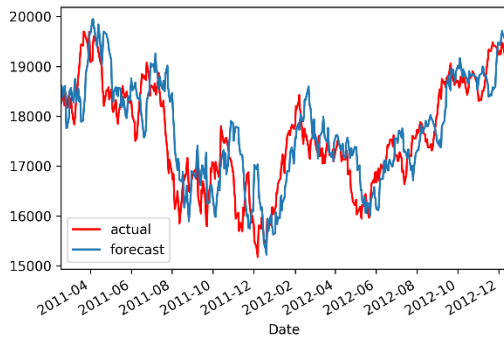
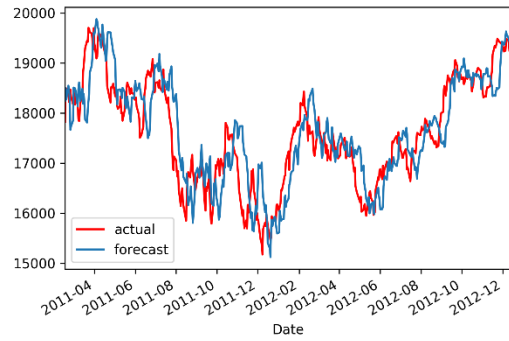
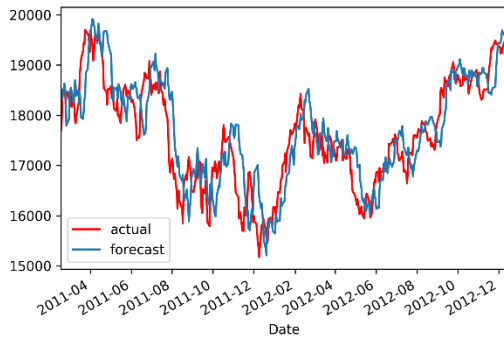
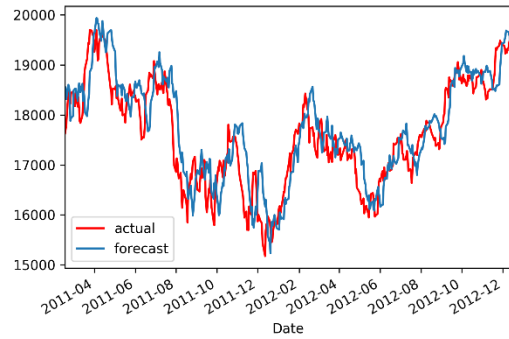
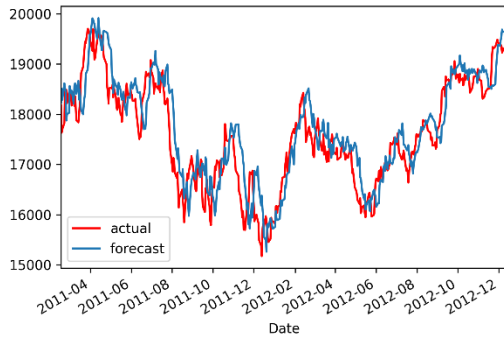
LSTM Results:



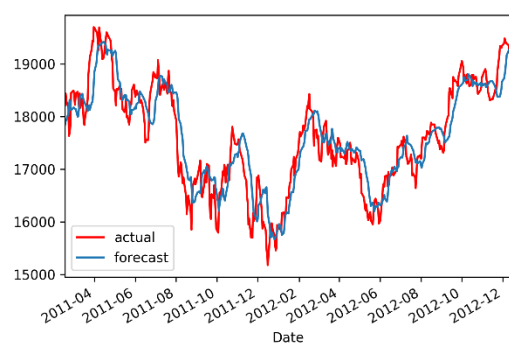
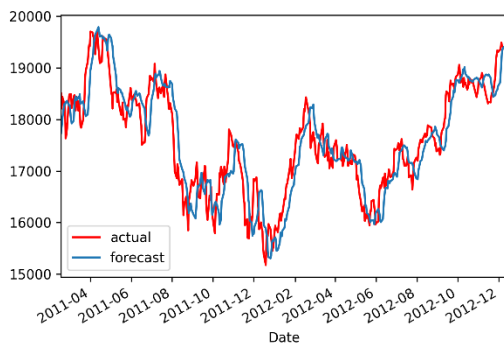


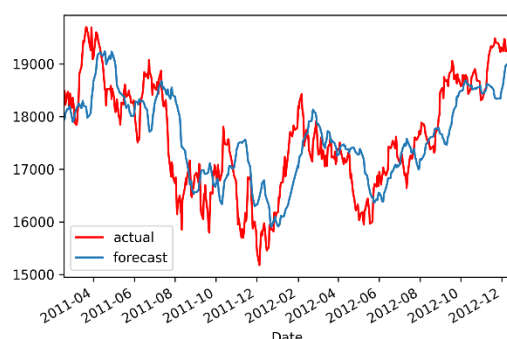
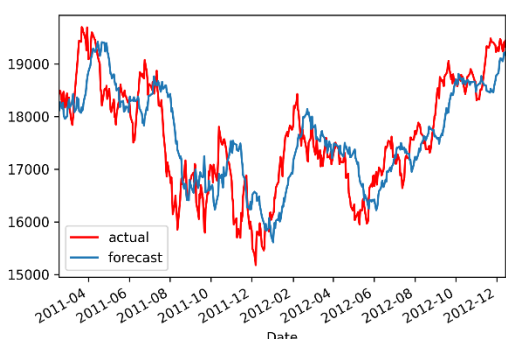
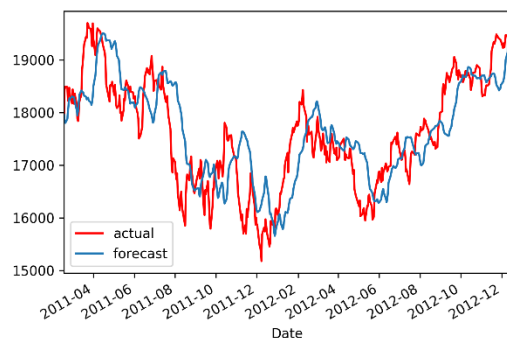
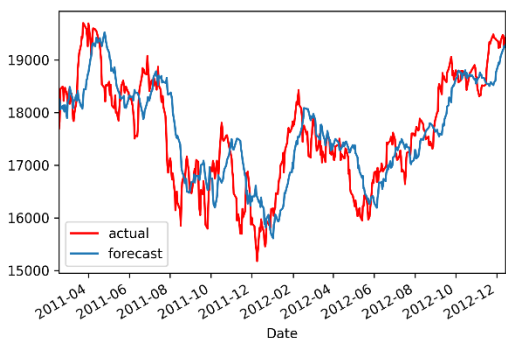
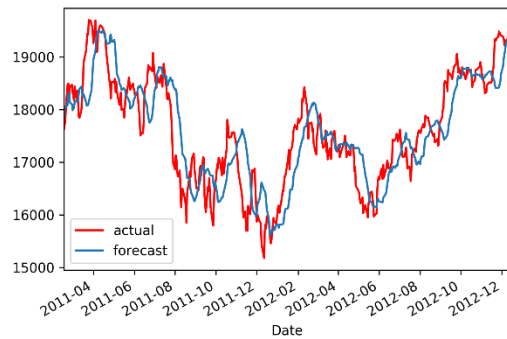
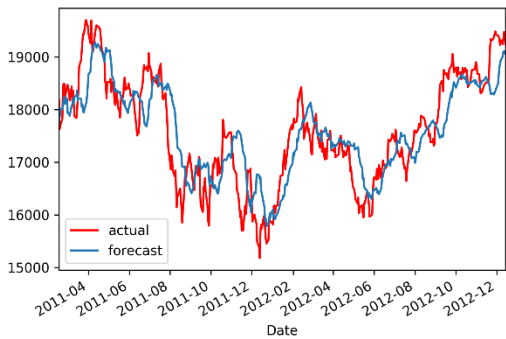
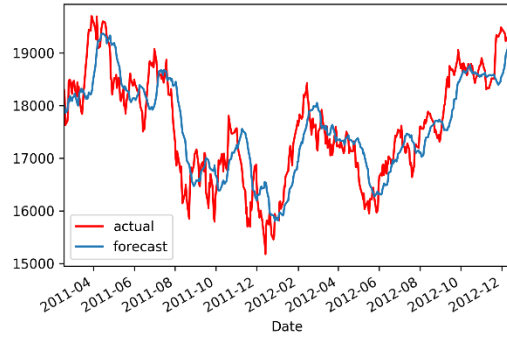
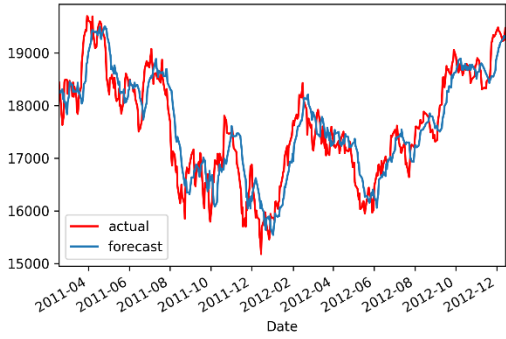
LinearSVR Results:



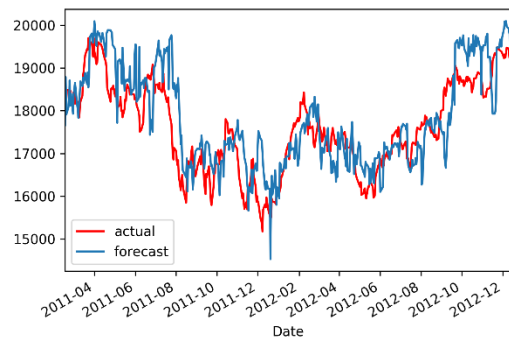
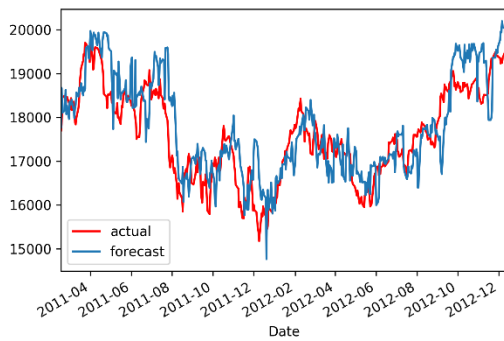
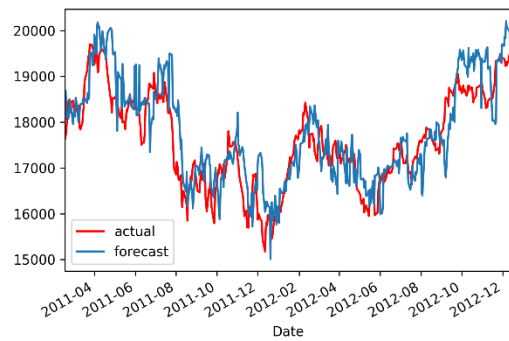
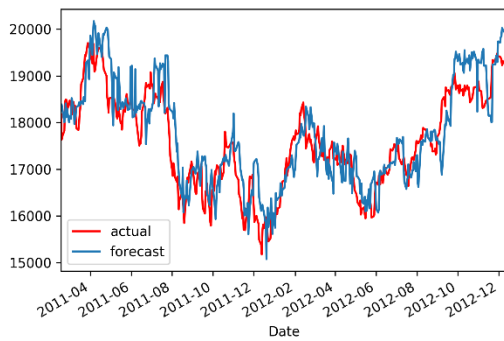
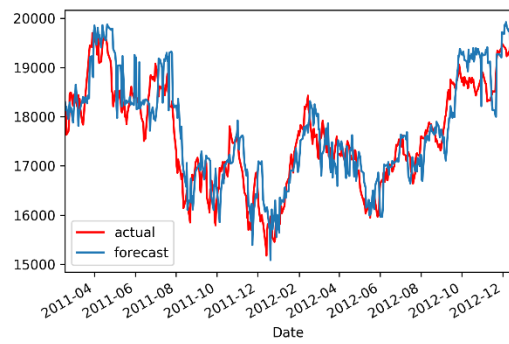
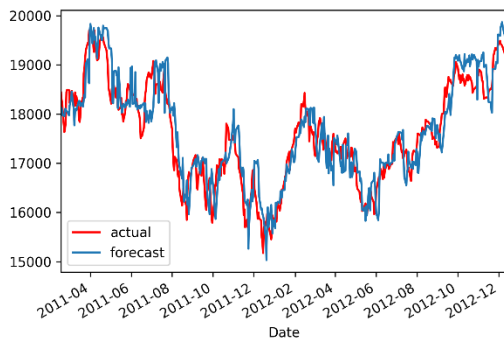
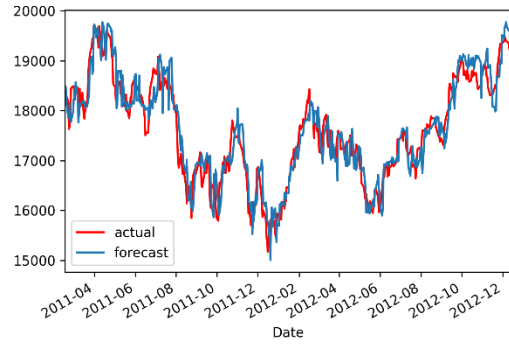
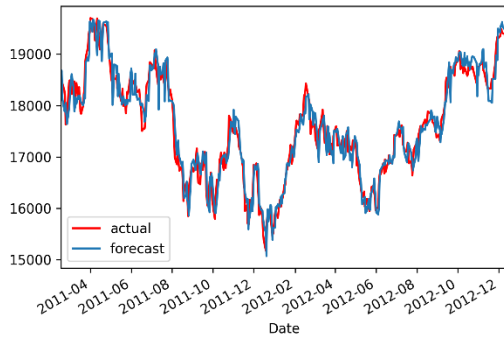


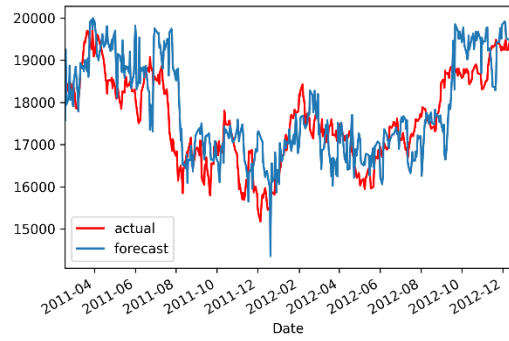
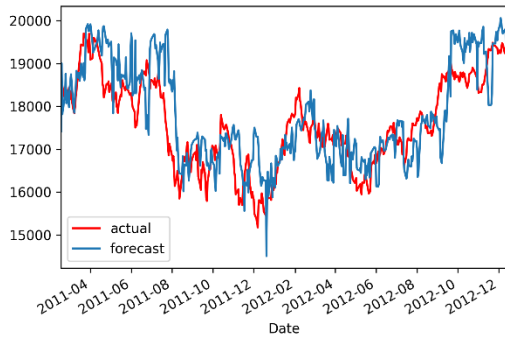
**MLP Results:**



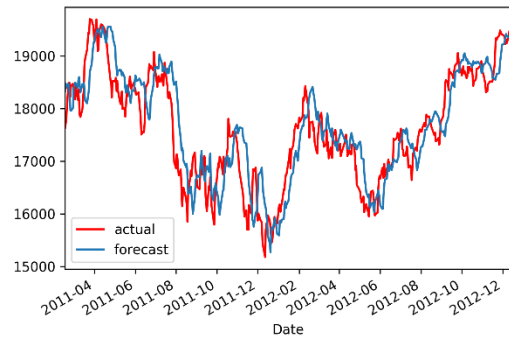
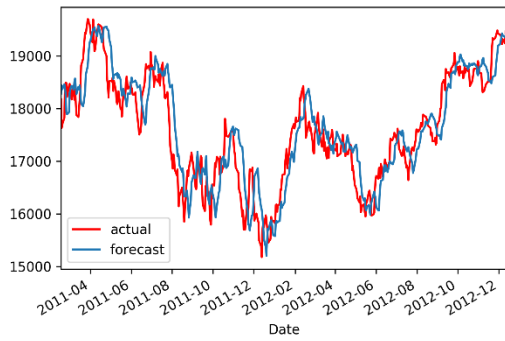
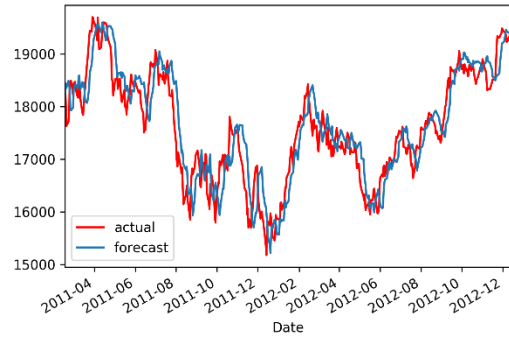
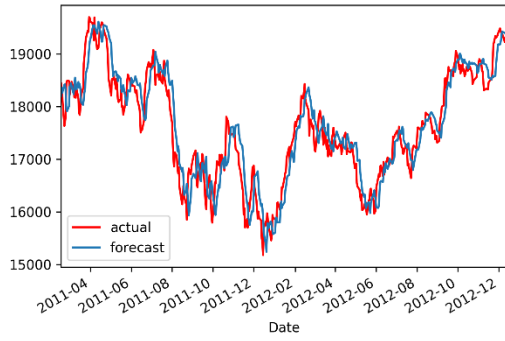
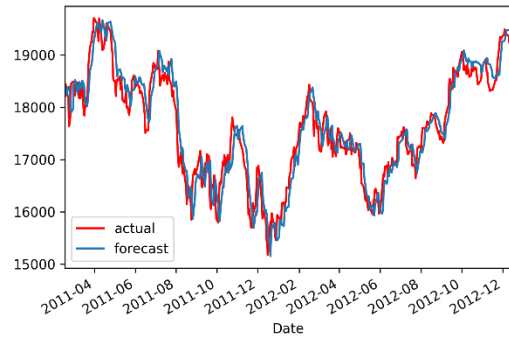
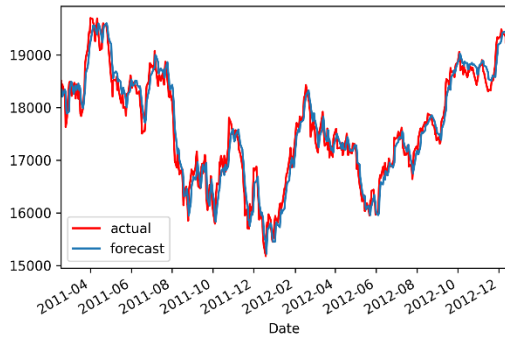


RN Results:

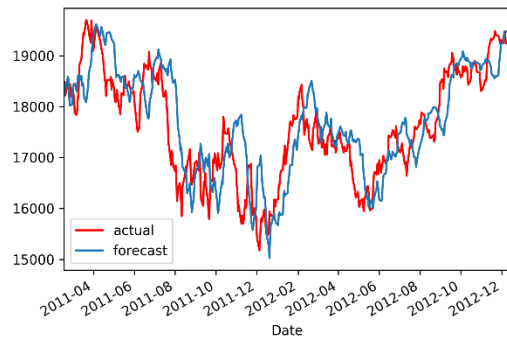
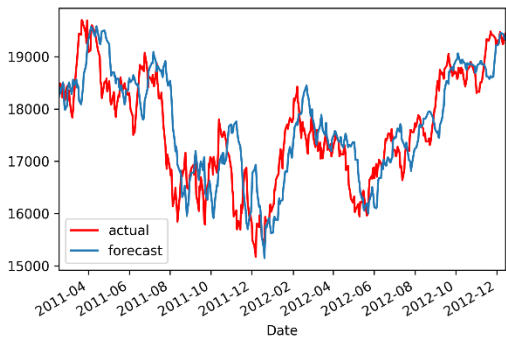
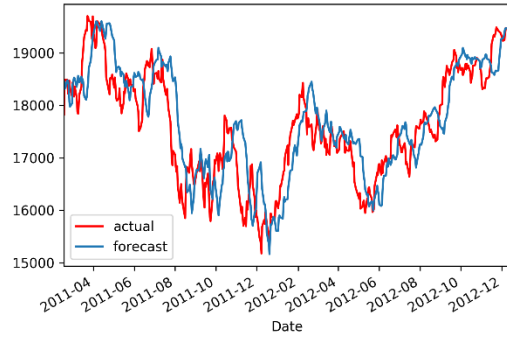
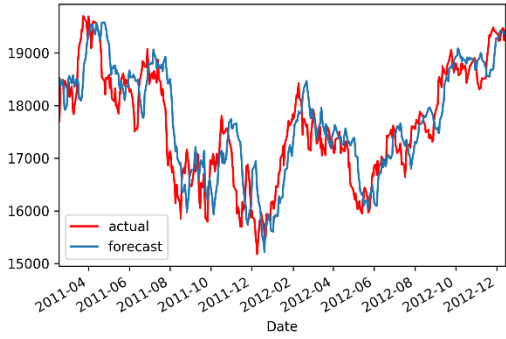




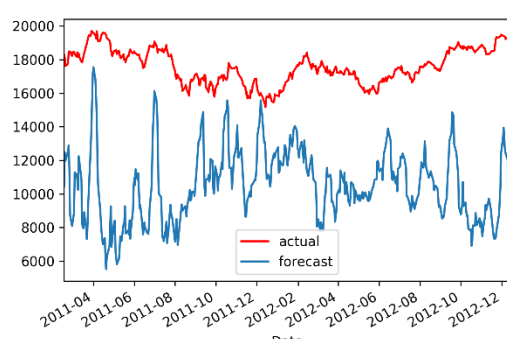
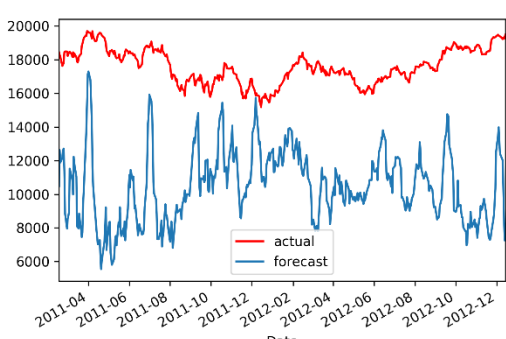
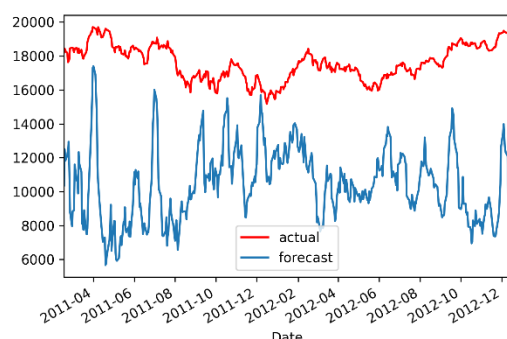
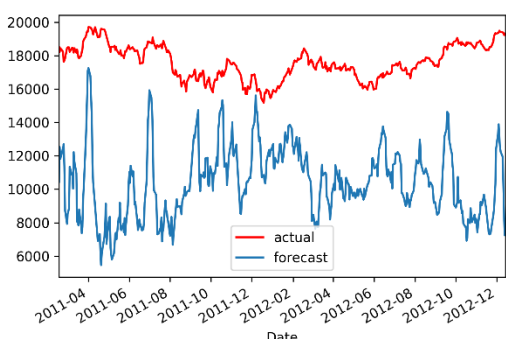
**SVR Results:**

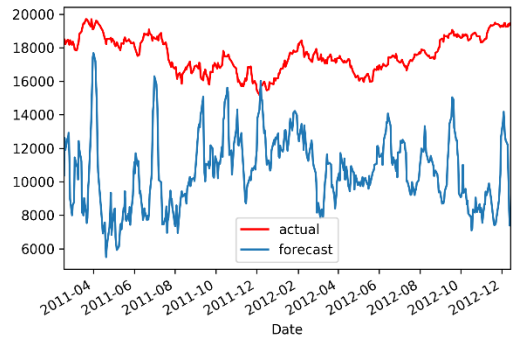
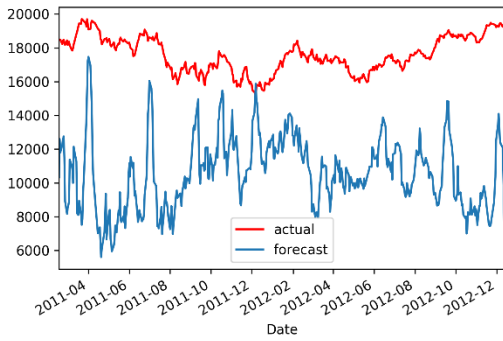
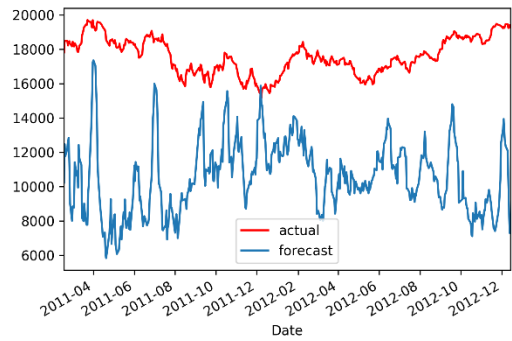
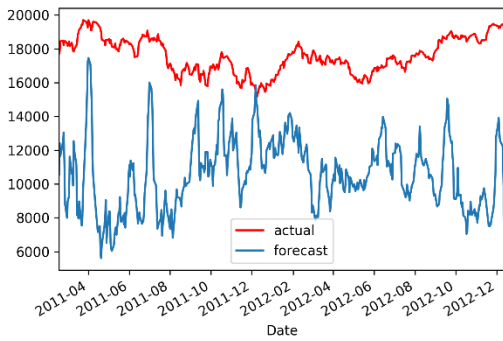
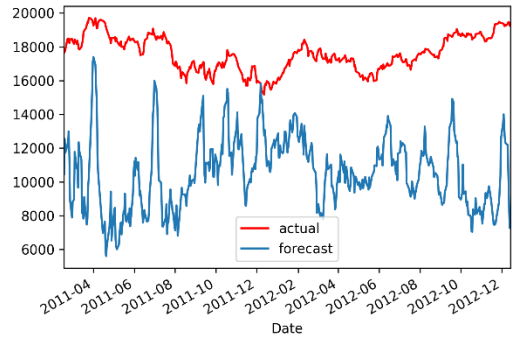
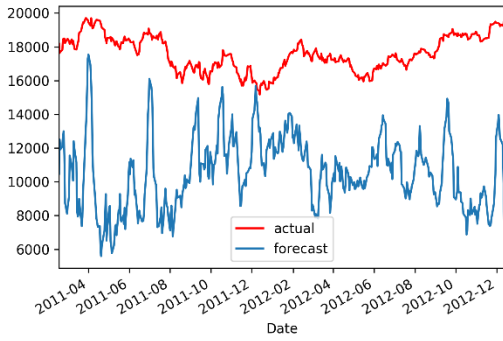




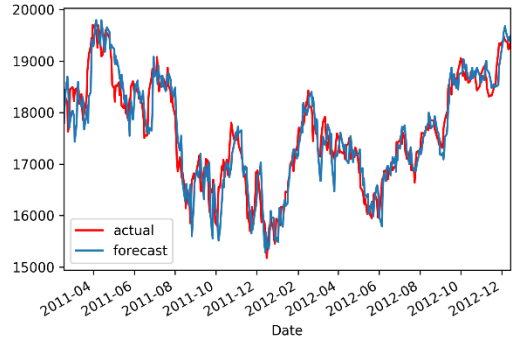
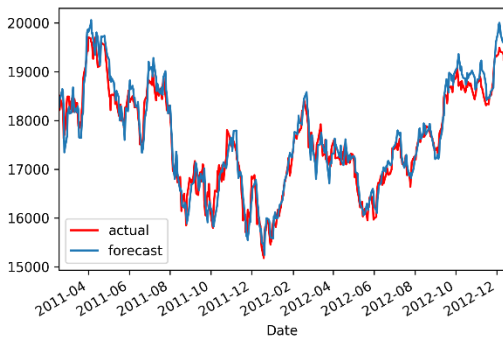


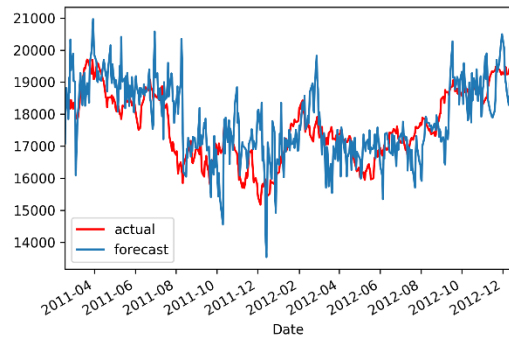
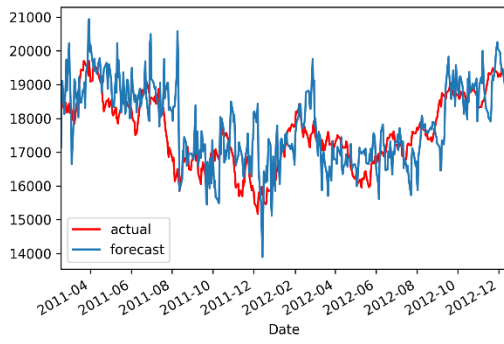
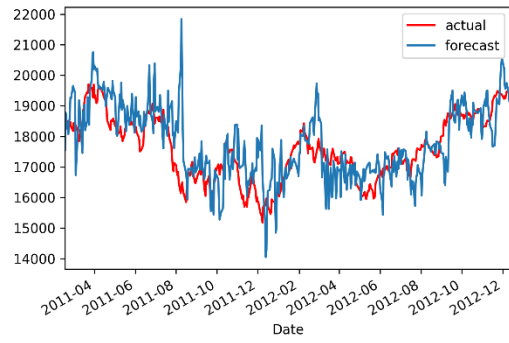
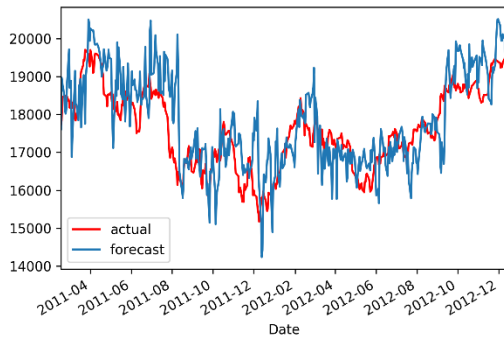
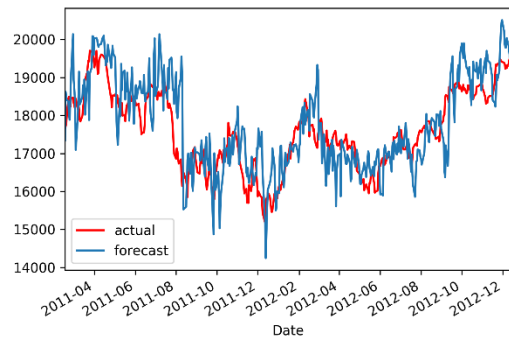
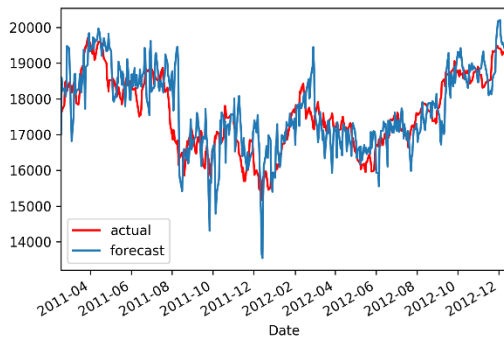
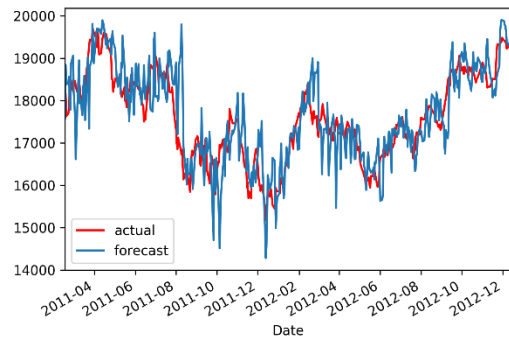
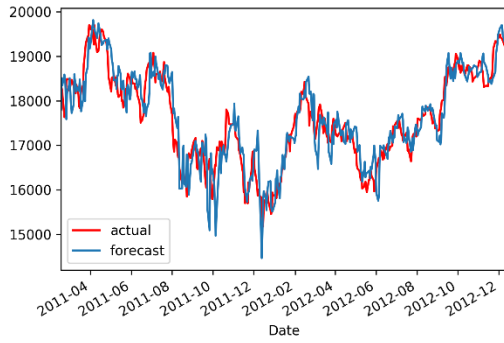
**SVR-ply Results:**





**SVR-RBF Results:**





## SVR-RBF 2 Results:

