FORECASTING STOCK PRICE VIA VANILLA LSTM AND BI-DIRECTIONAL LSTM

SAURABH SINGH

FINAL THESIS REPORT

JUNE 2022

**TABLE OF CONTENTS**

ACKNOWLEDGEMENTS 4

ABSTRACT 5

LIST OF TABLES 6

LIST OF FIGURES 7

LIST OF ABBREVIATIONS 9

CHAPTER 1: INTRODUCTION 10

1.1 Background 10

1.2 Problem Statement 10

1.3 Aim and Objectives 11

1.4 Research Question 12

1.5 Scope of the Study 12

1.6 Significance of the Study 12

CHAPTER 2: LITERATURE REVIEW 13

2.1 Introduction 13

2.2 Time Series Forecasting 13

2.3 Support Vector Machines 16

2.4 Neural Networks 16

2.4.1 Artificial Neural Network 16

2.4.2 Convolutional Neural Network 18

2.4.3 Recurrent Neural Network 24

2.4.4 Long Short-Term Memory Network 26

2.4.4.1 Vanilla LSTM 29

2.4.4.2 Bi-Directional LSTM 31

2.5 Discussion 32

2.6 Summary 34

CHAPTER 3: RESEARCH METHODOLOGY 35

3.1 Introduction 35

3.2 Research Approach 35

3.2.1 Data Selection 36

3.2.2 Data Pre-processing 37

3.2.2.1 Missing Value Treatment 37

3.2.2.2 Feature Elimination 37

3.2.2.3 Exploratory Data Analysis 38

3.2.2.4 Data Visualization 38

3.2.2.5 Feature Scaling 39

3.2.2.6 Train-Test Split 39

3.2.3 Model Development 40

3.2.3.1 Vanilla LSTM 40

3.2.3.2 Bi-Directional LSTM 40

3.2.4 Model Evaluation 41

3.3 Summary 41

CHAPTER 4: ANALYSIS AND DESIGN 42

4.1 Introduction 42

4.2 Dataset Description 42

4.3 Dataset Preparation 42

4.3.1 Missing Value Treatment 43

4.3.2 Feature Elimination 43

4.3.3 Univariate Analysis 43

4.3.4 Data Visualization 45

4.3.5 Feature Engineering 47

4.3.6 Feature Scaling 47

4.3.7 Train-Test Split 48

4.4 Model Development and Implementaion 48

4.4.1 Vanilla LSTM 49

4.4.2 Bi-Directional LSTM 53

4.5 Summary 56

CHAPTER 5: RESULTS AND DISCUSSIONS 57

5.1 Introduction 57

5.2 Interpretations of Visualizations 57

5.3 Model Evaluation 57

5.3.1 Vanilla LSTM 58

5.3.2 Bi-Directional LSTM 58

5.4 Visualizations for the Optimal Model 59

5.5 Discussions 63

5.6 Summary 63

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS 64

6.1 Introduction 64

6.2 Discussions and Conclusions 64

6.3 Contribution to Knowledge 65

6.4 Future Recommendations 65

REFERENCES 66

APPENDIX B: RESEARCH PROPOSAL 69

**ACKNOWLEDGEMENTS**

This thesis is dedicated to my beloved parents who have been my constant source of inspiration. They have given me the drive and discipline to tackle any task with enthusiasm and determination.

I also dedicate this thesis to my thesis supervisor who never failed to teach and guide me, to my family and to my friends.

**ABSTRACT**

The stock market is considered to be dynamic and complex in nature. Predicting the future stock price is a challenging task as there are many factors involved making the stock price labile. These factors can be physical or psychological. Throughout these years various approaches have been put forward to predict the stock price including regression models, time series forecasting, support vector machines and neural networks. Among various neural networks, long short-term memory (LSTM) models can dispense state of the art predictions with proper adjustments of the hyper parameters. This research proposes a comparative study between a Vanilla LSTM model and a Bi-Directional LSTM model and highlights the advantages of hybrid predictive models over standalone deep learning models. The models are assessed on a freely accessible dataset from yahoo finance including the historical stock prices of Tata Motors, HDFC, and Reliance of National Stock Exchange (NSE) of India. The performance metric used is mean absolute percentage error (MAPE).

**LIST OF TABLES**

Table 2.1 MAPE values for the respective models (Rotela Junior, 2014)

Table 2.2 Performance metrics used for the models (Wang, 2020)

Table 2.3 Studies following hybrid approach in Predictive Modeling

Table 4.1 Execution time and MAPE for respective stocks (Vanilla LSTM)

Table 4.2 Execution time and MAPE for respective stocks (Bi-LSTM)

Table 5.1 Various Vanilla LSTM models for Open Price

Table 5.2 Various Vanilla LSTM models for Close Price

Table 5.3 Various Bi-Directional LSTM models for Open Price

Table 5.4 Various Bi-Directional LSTM models for Close Price

**LIST OF FIGURES**

Figure 2.1 Trend and Seasonal components of a time series

Figure 2.2 Predictions of ARIMA, Neural Network and hybrid model (Zhang, 2003)

Figure 2.3 Artificial Neuron (Menon, 2018)

Figure 2.4 Structure of an ANN (Kim Soon, 2013)

Figure 2.5 Structure diagram of CNN-LSTM model (Wang, 2020)

Figure 2.6 Predictions of the MLP model (Wang, 2020)

Figure 2.7 Predictions of the CNN model (Wang, 2020)

Figure 2.8 Predictions of the RNN model (Wang, 2020)

Figure 2.9 Predictions of the LSTM model (Wang, 2020)

Figure 2.10 Predictions of the CNN-RNN model (Wang, 2020)

Figure 2.11 Predictions of the CNN-LSTM model (Wang, 2020)

Figure 2.12 Structure of the proposed multiple pipeline model (Verma, 2019)

Figure 2.13 Structure of the CNN-RNN model (Wang, 2018)

Figure 2.14 MSEs of different forecasting approaches (Wang, 2018)

Figure 2.15 Structure of an LSTM memory block (Baek, 2018)

Figure 2.16 Structure of the proposed ModAugNet framework (Baek, 2018)

Figure 2.17 Vanilla LSTM architecture (Yuan, 2018)

Figure 2.18 Bi-Directional LSTM Architecture

Figure 2.19 Architecture of the multiple pipeline model (Eapen, 2019)

Figure 3.1 Research Approach

Figure 4.1 Boxplot of Open Price of Tata Motors

Figure 4.2 Boxplot of Close Price of Tata Motors

Figure 4.3 Boxplot of Open Price of HDFC

Figure 4.4 Boxplot of Close Price of HDFC

Figure 4.5 Boxplot of Open Price of Reliance

Figure 4.6 Boxplot of Close Price of Reliance

Figure 4.7 Open and Close Price of Tata Motors

Figure 4.8 Open and Close Price of HDFC

Figure 4.9 Open and Close Price of Reliance

Figure 4.10 Actual v/s Predicted for Open Price of Tata Motors (Vanilla LSTM)

Figure 4.11 Actual v/s Predicted for Open Price of HDFC (Vanilla LSTM)

Figure 4.12 Actual v/s Predicted for Open Price of Reliance (Vanilla LSTM)

Figure 4.13 Actual v/s Predicted for Close Price of Tata Motors (Vanilla LSTM)

Figure 4.14 Actual v/s Predicted for Close Price of HDFC (Vanilla LSTM)

Figure 4.15 Actual v/s Predicted for Close Price of Reliance (Vanilla LSTM)

Figure 4.16 Actual v/s Predicted for Open Price of Tata Motors (Bi-Directional LSTM)

Figure 4.17 Actual v/s Predicted for Open Price of HDFC (Bi-Directional LSTM)

Figure 4.18 Actual v/s Predicted for Open Price of Reliance (Bi-Directional LSTM)

Figure 4.19 Actual v/s Predicted for Close Price of Tata Motors (Bi-Directional LSTM)

Figure 4.20 Actual v/s Predicted for Close Price of HDFC (Bi-Directional LSTM)

Figure 4.21 Actual v/s Predicted for Close Price of Reliance (Bi-Directional LSTM)

Figure 5.1 Actual v/s Predicted for Open Price of Tata Motors (Optimal Model)

Figure 5.2 Actual v/s Predicted for Open Price of HDFC (Optimal Model)

Figure 5.3 Actual v/s Predicted for Open Price of Reliance (Optimal Model)

Figure 5.4 Actual v/s Predicted for Close Price of Tata Motors (Optimal Model)

Figure 5.5 Actual v/s Predicted for Close Price of HDFC (Optimal Model)

Figure 5.6 Actual v/s Predicted for Close Price of Reliance (Optimal Model)

**LIST OF ABBREVIATIONS**

LSTM Long short-term memory

ANN Artificial neural network

RNN Recurrent neural network

SVM Support vector machine

NN Neural network

CRNN Convolutional neural network

RMSE Root mean squared error

MAPE Mean absolute percentage error

NSE National Stock Exchange

DNN Deep Neural Network

**CHAPTER 1**

**INTRODUCTION**

* 1. **Background**

Investing is an act of committing money or capital now in the expectation of gaining more money in the future. Investing can be a means to become financially independent or it can help in preserving the accumulated wealth. Investing in the stock market is probably the most common way to invest, because of its advantages and charm.

There are stock exchanges all around the world. Investors or traders connect with the stock exchanges via their brokers to place buys or sell shares. Stock price prediction is a step further in making a valuable investment. While there are people that think that it might not be possible to predict future stock prices, however in the past years, researchers have evidently proved that it is possible to forecast future stock prices with high accuracy (Mehtab, 2020). More and more papers are being published with various methods of prediction involving regression models, time series forecasting and deep learning models. Among these methods, the most standardly utilized method is Artificial Neural Network (ANN) (Maswood 2020). ANNs are composed of a collection of artificial neurons, which loosely replicate the neurons in a biological brain. Although ANNs have shown acceptable results as they are capable of learning any nonlinear function, they do suffer from an over-fitting issue. However, Recurrent Neural Network (RNN) and especially LSTM models have an advantage over ANNs as they are capable of grasping long term dependencies in data. LSTM models have dispensed state of the art predictions with proper hyper-parameter tuning. This study proposes a prediction framework that can forecast future stock prices utilizing two LSTM models; Vanilla LSTM and Bi-Directional LSTM.

* 1. **Problem Statement**

Financial time series data has always been more complex and prone to errors than any other statistical data. There are two traditional approaches to forecast future stock prices; Technical analysis and Fundamental analysis (Kim Soon, 2013). Fundamental analysis uses revenues, future deals, profit margins and other factors to forecast future stock prices, whereas, technical analysis can be defined as a time series analysis to forecast future stock prices using the historical stock price data. Through the years, combination machine learning models such as SVMs, ANNs and RNNs have revealed complex non-linear patterns that are impossible to detect with linear algorithms (Moghar, 2020). These models have proved to be more effective in predicting the closing stock price than the linear regression models.

A feedforward neural network is the most elementary model used for stock price prediction (Kim Soon, 2013). The feedforward NN was the most fundamental type of ANN devised. Herein connections between the nodes do not form a cycle. A feedforward NN was used by (Morris, 2007) to predict the stock market index trading signals.

RNNs are considered to be good at modelling and processing sequential data, which is quite suitable for stock price prediction. RNN networks are popularly used on financial time series data for making predictions. RNNs take information from prior inputs to influence the current input and output. A CRNN forecasting model was proposed by (Wang, 2018) to predict future stock price of 9 forex pairs.

A LSTM model is a special kind of RNN that overcomes the problem of exploding gradient which occurs in traditional RNNs (Ma, 2020). LSTMs are capable of learning long term dependencies in time series data. A basic LSTM architecture consists of :

* a cell
* an input gate
* an output gate
* a forget gate

A comparative study of LSTM and Bi-Directional LSTM model was published by (Maswood, 2020), proposing that Bi-Directional LSTM yields lower root mean squared error (RMSE) than a traditional LSTM.

* 1. **Aim and Objectives**

This research aims to propose a stock price prediction framework that can forecast future stock prices with proper accuracy. The goal of this study is to forecast the future stock prices using a Vanilla LSTM and a Bi-Directional LSTM.

The research objectives are as follows:

* To study state of the art prediction frameworks used for forecasting future stock prices.
* To compare the Vanilla LSTM and Bi-Directional LSTM models by varying the number of epochs, hidden layers, dense layers and other units.
* To evaluate the performance of the proposed models based on appropriate error metrics.
  1. **Research Question**

This research aims to propose a stock price prediction framework that can forecast future stock prices with proper accuracy. The goal of this study is to forecast the future stock prices using a Vanilla LSTM and a Bi-Directional LSTM.

* 1. **Scope of the Study**

This study proposes a prediction framework based on the Vanilla LSTM and Bi-Directional LSTM models to forecast future stock prices of NSE of India. The models are to be assessed upon the historical stock price data of Tata Motors, HDFC and Reliance stocks having a time span of 6 months.

* 1. **Significance of the Study**

The motivation behind this research comes from the work at Bennett Coleman and Co. Ltd. (Times Group). From examining business performance to spearheading market share analysis, day to day work also includes forecasting monthly revenue projections across various business segments that helps the sales team to push the de-growing agents. This study can be used in future literature review involving Vanilla LSTM and Bi-Directional LSTM.

**CHAPTER 2**

**LITRATURE REVIEW**

**2.1 Introduction**

This chapter reviews the studies carried out on future stock price prediction frameworks. In the field of future stock price prediction, various linear and nonlinear approaches have been put forward to predict future stock price including regression models, time series forecasting, support vector machines and neural networks. This chapter begins with traditional time series forecasting techniques followed by a detailed review of the prominent deep learning models that have proved to dispense state of the art predictions. Some extensively used deep learning models are Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This chapter highlights the advantages of a hybrid model comprising of multiple deep learning models over traditional deep learning models.

**2.2 Time Series Forecasting**

Forecasting can be defined as the prediction of some future event or events by analysing the historical data. Forecasting spans many fields such as business, economics, finance and weather. Usually forecasting can be grouped into qualitative forecasting and quantitative forecasting. While qualitative forecasting relies upon human intellect, quantitative forecasting is based on historical data or time series as quantitative forecasting assumes that past observations or values are relevant for predicting the future. A time series is a sequence of observations taken sequentially in time. Time series forecasting is a quantitative forecasting method used for forecasting or predicting future values over a period of time. Time series forecasting is basically looking at the past data to make predictions into the future. Some examples of time series forecasting are sales forecasting, weather forecasting and demand forecasting.

A popular use case of time series forecasting is stock price prediction. Exponential smoothing and Autoregressive Integrated Moving Average (ARIMA) models are the two most extensively used time series forecasting techniques. While exponential smoothing techniques, such as Holt-Winters’s exponential smoothing technique, better capture the trend and the seasonality of the time series as illustrated in figure 2.1, ARIMA models aim to describe the auto-correlation in the data. In exponential smoothing techniques the past observations are weighted in an exponentially decreasing order, meaning most recent observations are given higher weights than far-away values whereas in ARIMA models a moving average takes an average of past values and weigh them equally. In (Zhang, 2003), the author has proposed that ARIMA models are particularly useful when there is no information available of the time series data.

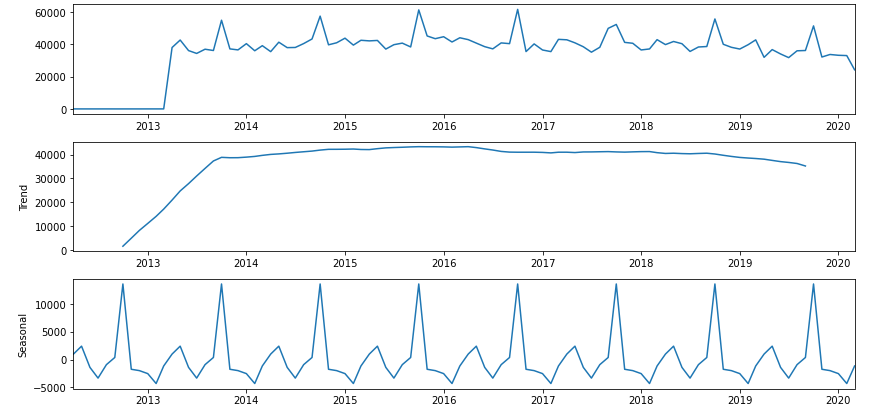


Figure 2.1 Trend and Seasonal components of a time series

However, recent studies in the field of stock price prediction may imply that artificial neural networks can be a promising alternative to the traditional linear models because of their nonlinear modelling capability. To address this issue (Zhang, 2003) proposed a combining approach with a linear ARIMA model and a nonlinear ANN model, aiming to capture the unique strength of linear and nonlinear modelling. This combined approach can be effective against complex problems having linear and nonlinear correlation structures. In (Zhang, 2003), the author has proposed that the hybrid model has outperformed the isolated models as illustrated in the figure 2.2.

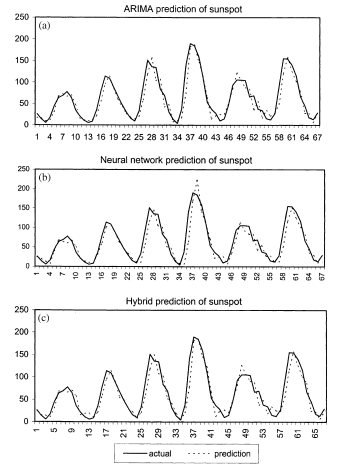


Figure 2.2 Predictions of ARIMA, Neural Network and hybrid model (Zhang, 2003)

Similar to (Zhang, 2003), the authors in (Prapanna, 2014) conducted a study on the effectiveness of ARIMA model on 56 Indian stocks from different sectors. In another approach (Rotela Junior, 2014), the authors evaluated the performance of an ARIMA model for future stock price predictions of Ibovespa. The research approach followed the popular Box-Jenkins method with MAPE as the performance metric. The authors concluded that the ARIMA model outperformed the single exponential smoothing model and the double exponential smoothing model. The MAPE values for the respective models are highlighted in table 2.1.

Table 2.1 MAPE values for the respective models (Rotela Junior, 2014)

|  |  |  |  |
| --- | --- | --- | --- |
|  | ARIMA | Single Exponential | Double Exponential |
| MAPE | 0.052 | 0.086 | 0.118 |

**2.3 Support Vector Machines**

Support vector machines are supervised learning models that are particularly used for classification and regression analysis. Recently SVMs are commonly being used for future stock price prediction. SVMs are linear learning models that use a linear function. The best line is said to be the one that minimises the cost function (Thissen, 2003) expressed in equation 2.1.

(2.1)

The initial part of the cost function is a weight decay which is used to regularize weight sizes and penalizes large weights because of which the weights converge to smaller values. Large weights diminish the generalization of SVMs as they cause high variance. The second part is a penalty function that penalizes errors larger than. The prediction performance of SVMs are sensitive to these parameters, thus it is important to find the optimal value of these parameters. In (Kim, 2003), the author has compared SVM with back propagation neural network (BPN) and case-based reasoning (CBR). The results showed that the SVM surpassed BPN and CBR. Similar to (Kim, 2003), the authors in (Tay, 2001) have examined the predictability with SVMs against the BPN networks and found that SVMs provide a promising alternative to BPN networks.

**2.4 Neural Networks**

The application of deep learning models in forecasting future stock prices has been an appealing area of research for decades. Deep learning models have proved to be much more precise and accurate in predicting a market value close to the tangible value than the traditional forecasting techniques. This dominance can be attributed to the use of various neural networks such as ANNs, CNNs, RNNs, LSTM and their hybrid models.

**2.4.1 Artificial Neural Network**

Artificial neural networks (ANNs), usually called neural networks, are computational structures that act in a similar manner to that of a biological neural network that constitute animal brains. The authors in (Menon, 2018) proposed ANNs to be a nonlinear statistical data tool that can function the intricate relation between the past values. Being good function approximators, ANNs can learn the underlying patterns from the past or historic data and make accurate predictions.

An ANN is a collection of connected nodes called artificial neurons, which are inspired by the neurons in a biological brain. An artificial neuron receives inputs or signals from other neurons with associated weights and produces an output using a nonlinear activation function. An activation function is used to control the firing of the neuron. An artificial neuron with m inputs is illustrated in figure 2.3.

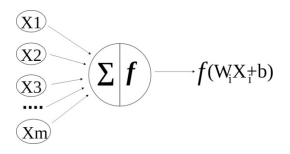


Figure 2.3 Artificial Neuron (Menon, 2018)

An ANN consists of vertically stacked components called layers. There are 3 types of layers present in an ANN:

* input layer
* hidden layer
* output layer

The input layer accepts the data and passes it to the rest of the network. There can be any number of hidden layers present in an ANN, making the ANN more complex. The output layer holds the output of the network. The structure of a typical ANN is illustrated in figure 2.4.

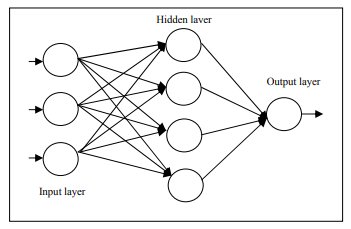


Figure 2.4 Structure of an ANN (Kim Soon, 2013)

A feedforward neural network (Kim Soon, 2013) is the most elementary model used for forecasting future stock price. The feedforward neural network is the most fundamental type of ANN devised where the information only flows in forward direction. Herein connection between the nodes do not form a cycle. The information flow in a feedforward NN is from the input layer, crossing the hidden layers to the output layer. In (White, 2002), the author used a feedforward NN to predict the IBM daily common stock price. Similar to (White, 2002), the authors in (Morris, 2007) used a feedforward NN to predict the stock market index trading signals. In (Yetis, 2014), the authors made use of a feedforward NN to forecast NASDAQ’s stock value with given input parameter of stock market.

**2.4.2 Convolutional Neural Network**

A convolutional neural network (CNN) is a type of artificial neural network that is specifically designed to process pixel data. CNN has become dominant in various computer vision tasks, particularly used in image recognition and processing, however, it can be effectively applied to forecast future stock prices. A CNN is composed of three layers:

* convolution layer
* pooling layer
* fully connected layer

The convolution layer is the first layer of a convolutional neural network that can be followed by additional convolution layers or pooling layers. The network ends at a fully connected layer. With each increasing layer, the complexity of the CNN increases. While the early layers learn simple features such as colours and edges, the final layers recognise larger elements such as the shape of the object. The convolution layer (Tai Wu, 2021) contains several convolution kernels composed of cells, where each cell has a bias and weight associated to it similar to an artificial neuron in a feedforward neural network, whose function is to collect the features of the input data. This process is known as convolution. Pooling layer reduces the number of parameters by conducting dimensionality reduction. Here the kernel does not have any weights, instead, the kernel applies an aggregate function to the values inside the receptive field. The fully connected layer classifies the images based on the features extracted through the previous layers. While the convolution layers use rectified linear unit (ReLU) as the activation function, fully connected layers use a softmax activation function to classify the images by returning a probability from 0 to 1.

Though the neural networks have obvious advantages in nonlinear data forecasting, their accuracy can still be improved. The authors in (Wang, 2020) proposed that the forecasting accuracy of CNN alone is relatively low against the hybrid models. In (Wang, 2020), the authors proposed a combined approach of CNN and LSTM to forecast future stock price. Here the CNN was used to extract the time feature of data and the LSTM was used to learn the extracted feature data and forecast the future stock price. The structure diagram of the CNN-LSTM based model is illustrated in figure 2.5.

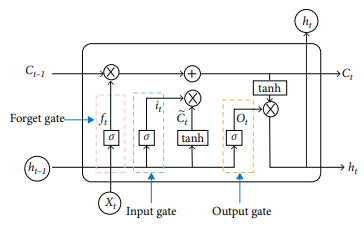


Figure 2.5 Structure diagram of CNN-LSTM model (Wang, 2020)

The authors (Wang, 2020) proposed that the CNN-LSTM based model has the highest accuracy compared to the MLP, CNN, RNN, LSTM and CNN-RNN models with MAE and RMSE as the performance metrics. The respective time series plots are illustrated in figure 2.6, 2.7, 2.8, 2.9, 2.10 and 2.11.

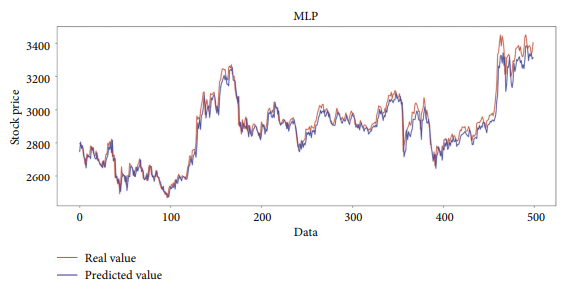


Figure 2.6 Predictions of the MLP model (Wang, 2020)

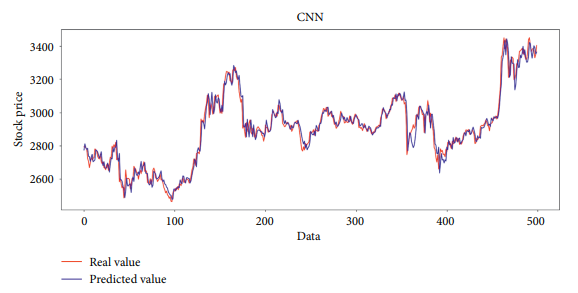


Figure 2.7 Predictions of the CNN model (Wang, 2020)

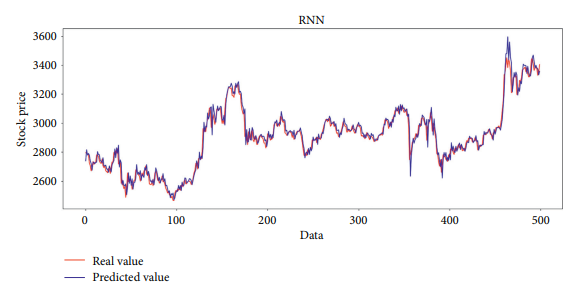


Figure 2.8 Predictions of the RNN model (Wang, 2020)

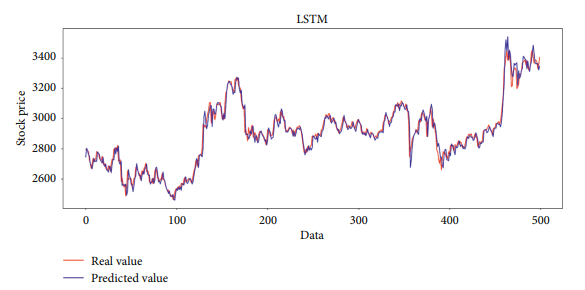


Figure 2.9 Predictions of the LSTM model (Wang, 2020)

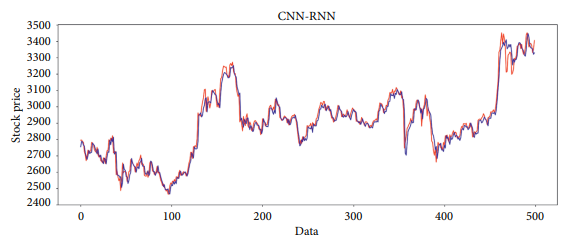


Figure 2.10 Predictions of the CNN-RNN model (Wang, 2020)

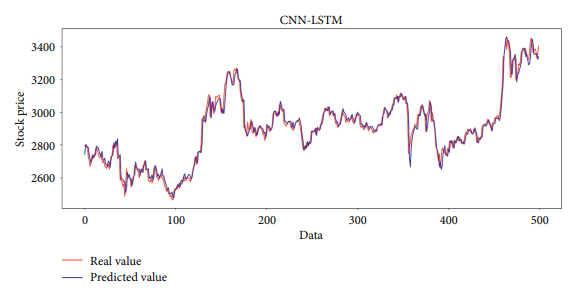


Figure 2.11 Predictions of the CNN-LSTM model (Wang, 2020)

The performance metrics used for the various approaches are highlighted in table 2.1.

Table 2.2 Performance metrics used for the models (Wang, 2020)

|  |  |  |  |
| --- | --- | --- | --- |
| Method | MAE | RMSE |  |
| MLP | 37.584 | 49.799 | 0,9442 |
| CNN | 30.138 | 42.967 | 0.9585 |
| RNN | 29.916 | 42.957 | 0.9593 |
| LSTM | 28.712 | 41.003 | 0.9622 |
| CNN-RNN | 28.285 | 40.538 | 0.9630 |
| CNN-LSTM | 27.564 | 39.688 | 0.9646 |

Another approach (Tai Wu, 2021) focused on a graph based CNN-LSTM stock price prediction algorithm and named the algorithm as stock sequence array convolutional LSTM (SACLSTM). A sequence array of historical data is fed to SACLSTM as the input image of the CNN framework. SACLSTM extracts certain features via the convolution layer and the pooling layer. The authors verified that the neural network framework when combined with CNN and LSTM units showed better performance than the traditional CNN and LSTM models.

Similar to (Tai Wu, 2021), the authors in (Eapen, 2019) proposed a novel deep learning approach combining multiple pipelines of CNN and Bi-Directional LSTM units. The proposed model was able to improve the prediction accuracy while minimizing the effects of overfitting by incorporating multiple pipelines of different CNN kernel sizes and Bi-Directional LSTM units. The structure of the proposed multiple pipeline model is illustrated in figure 2.12.

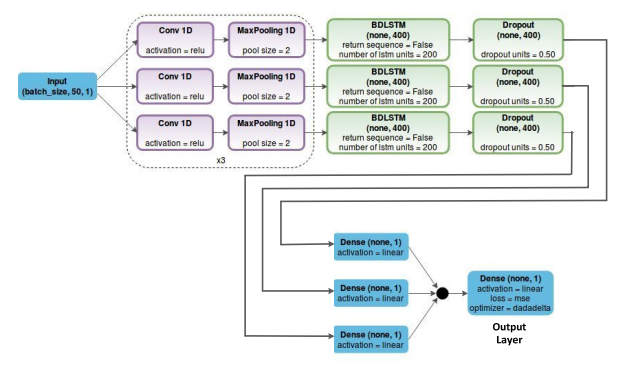


Figure 2.12 Structure of the proposed multiple pipeline model (Eapen, 2019)

In (Mehtab, 2020), the authors have proposed a univariate encoder-decoder convolutional LSTM model to predict future stock price. Among the other univariate CNN models, the univariate encoder-decoder convolutional LSTM was found to be the most accurate model, just not the fastest one.

**2.4.3 Recurrent Neural Network**

A recurrent neural network (RNN) is a special kind of neural network specifically designed to work with sequential data or time series data. The traditional feedforward neural networks are unable to process temporal information or sequential data such as a sentence, whereas, RNNs are designed for this very purpose. RNNs are considered to be good at modelling and processing sequential data, which is quite suitable for stock price prediction. Like a feedforward NN or a CNN, RNNs utilize training data to learn. An RNN keeps a memory of what it has already processed and thus can learn from previous iterations during its training. RNNs take information from prior inputs to influence the current input and output. While traditional NNs assume that inputs and outputs are independent of each other, the outputs of RNN depend on the previous elements within the sequence. RNNs are composed of three elementary components: the input layer, the hidden layers, and the output layer and each layer is composed of nodes. In RNNs, connections between the nodes form a directed circle. While a traditional feedforward NN has different weights associated to each node, RNNs share the same weight parameter within each layer of the network.

The authors in (Selvin, 2017) have compared CNN, RNN, and LSTM models to predict Infosys, TCS and Cipla stock prices. In another approach (Wang, 2018), the authors proposed a C-RNN forecasting framework based on a CNN and an RNN for Forex time series data. The structure of the proposed network is illustrated in figure 2.13.

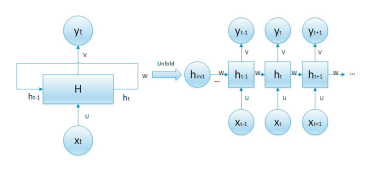


Figure 2.13 Structure of the CNN-RNN model (Wang, 2018)

Here is the input for RNN at time t, is the hidden state of the RNN at time t, is the output of the RNN at time t and u, v and w are the parameter matrices.

For any time instant t, the hidden state is calculated using the input of the present moment and the hidden state of the previous moment as expressed in equation 2.2.

(2.2)

Here is the activation function of the RNN and is the offset of the linear relationship. The output of the RNN at present moment t is expressed in equation 2.3.

(2.3)

Here is the offset of the linear relationship.

In (Wang, 2018), the authors have compared the C-RNN approach with traditional CNN and LSTM models and found the mean squared error (MSE) of the C-RNN approach for all nine exchange rate data of foreign exchange currency pairs. The comparison of MSEs of different forecasting approaches is illustrated in figure 2.14.

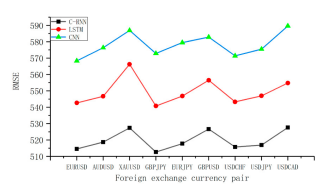


Figure 2.14 MSEs of different forecasting approaches (Wang, 2018)

RNNs specialise in learning temporal patterns from sequential data. RNNs remember the previous information and use it to process the current input and output. However, RNNs are not capable of learning long-term dependencies due to the popular vanishing gradient problem. This means that RNNs face difficulties in retaining for a long period of time. LSTMs are specifically designed to tackle the long-term dependency problem.

The authors in (Minami, 2018) proposed a hybrid sequential learning model using LSTM and RNN models. The authors incorporated the event information and the order backlog by the stock company. The authors concluded that event information may be effective in future stock price prediction.

**2.4.4 Long Short-Term Memory Network**

Long short-term memory (LSTM) networks are a special kind of RNNs that are capable of learning long-term dependencies allowing the information to persist. LSTMs are capable of handling the vanishing gradient problem that usually occurs in RNNs. LSTMs are specifically designed to tackle the long-term dependency problem. LSTMs have been widely used in speech recognition, time series analysis and text analysis. LSTMs have gating mechanism to regulate the flow of information. A basic LSTM architecture consists of:

* a cell
* an input gate
* an output gate
* a forget gate

The structure of an LSTM memory block (Baek, 2018) consisting of memory cells and gates is illustrated in figure 2.15.

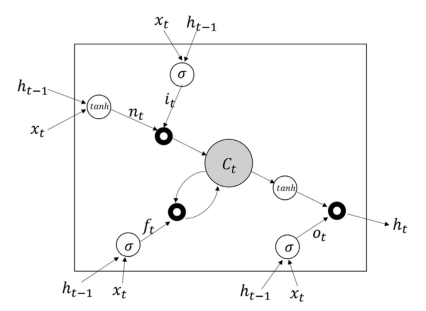


Figure 2.15 Structure of an LSTM memory block (Baek, 2018)

The authors in (Wang, 2020) have stated that the output of the previous moment and the input of the current moment are fed to the forget gate and the output of the forget gate is calculated as expressed in equation 2.4.

(2.4)

Here is the weight of the forget gate and is the bias of the forget gate, is the input of the current moment and is the output of the last moment.

The output of the previous moment and the input of the current moment are then fed into the input gate and the output and the candidate cell state of the input gate are calculated as expressed in equations 2.5 and 2.6 respectively.

(2.5)

(2.6)

Here is the weight of the input gate, is the bias of the input gate, is the weight of the candidate input gate and is the bias of the candidate input gate.

The current cell state is updated as expressed in equation 2.7.

(2.7)

The output and the input are then fed to the output gate at moment t and the output of the output gate is calculated as expressed in equation 2.8.

(2.8)

Here is the weight of the output gate and is the bias of the output gate.

The output of the LSTM network is calculated as expressed in equation 2.9.

(2.9)

The authors in (Baek, 2018) proposed ModAugNet framework consisting of two models: an overfitting prevention LSTM model and prediction LSTM model. The proposed framework was evaluated on S&P500 and KOSPI200 stock prices and confirmed an excellent prediction accuracy. The architecture of the proposed model is illustrated in figure 2.16.

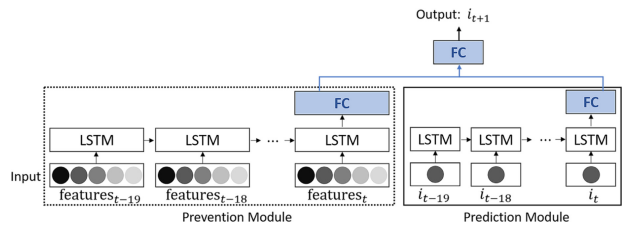


Figure 2.16 Structure of the proposed ModAugNet framework (Baek, 2018)

The authors (Baek, 2018) proposed that as the application of neural networks is a data-driven approach, deep neural networks require vast amount of data to avoid overfitting. An alternative could be data augmentation, however, it is not advised to augment a time series by transforming the original data as it would be artificial. Therefore, the authors proposed a modular forecasting framework for financial time series forecasting to make the model robust to overfitting.

In another approach (Zhang, 2020), the authors proposed a hybrid prediction framework by incorporating complementary ensemble empirical mode decomposition (CEEMD) as a sequence smoothing and decomposition module with LSTM network. The proposed model consists of CEEMD, PCA and LSTM. The purpose of CEEMD is to decompose the fluctuations of different scales from the time series data. The purpose of PCA is to perform data dimensionality reduction and extract the high level features which are then fed to the LSTM network to forecast the closing stock price. The proposed model outperformed the benchmark models in both predictive accuracy and profitability performance. The performance metrics used were RMSE, MAE and NMSE.

Similar to (Zhang, 2020), the authors in (Mehtab, 2019) proposed a hybrid approach for stock price prediction using machine learning, deep learning and natural language processing. The proposed model was evaluated on the NIFTY 50 index values of NSE of India over a period of three years. The LSTM network was augmented by integrating a sentiment analysis module to incorporate public sentiments regarding the stocks. The proposed model was tested on a cross validation technique based on self-organising fuzzy neural networks (SOFNN). The performance of the proposed model was found to be the best among all the traditional models proving that public sentiments can significantly impact the prediction accuracy.

The authors in (Yao, 2018) implemented an LSTM model for future stock price predictions of some randomly selected stocks from CSI 300 constituent stocks. The authors concluded that the precision, recall rate and the critical error of the LSTM model are better than that of random predictions. Similar to (Yao, 2018), the authors in (Wang, 2018) implemented and compared an LSTM model with a back propagation neural network (BPN). The LSTM model outperformed the BPN model in forecasting future stock prices with high accuracy. The authors in (Parmar, 2018) evaluated and compared an LSTM model with a regression model for future stock price prediction. The LSTM model outperformed the regression model.

The performance of an LSTM model varies with the choice of hyper parameters. Being a relatively new model, there are no established guidelines for configuring LSTM. To address this issue (Yadav, 2019) proposed a study on optimizing LSTM for future stock price prediction in Indian market. The recent works (Jin, 2020) proposed a deep learning-based stock market prediction model that considers investors’ emotional tendency by involving investors’ sentiments for future stock price prediction.

**2.4.4.1 Vanilla LSTM**

A Vanilla LSTM is a LSTM model that has a single hidden layer of LSTM units, and an output layer used to make a prediction. This variant of LSTM is less common in the literature. The illustration of Vanilla LSTM is illustrated in figure 2.17 (Yuan, 2018).

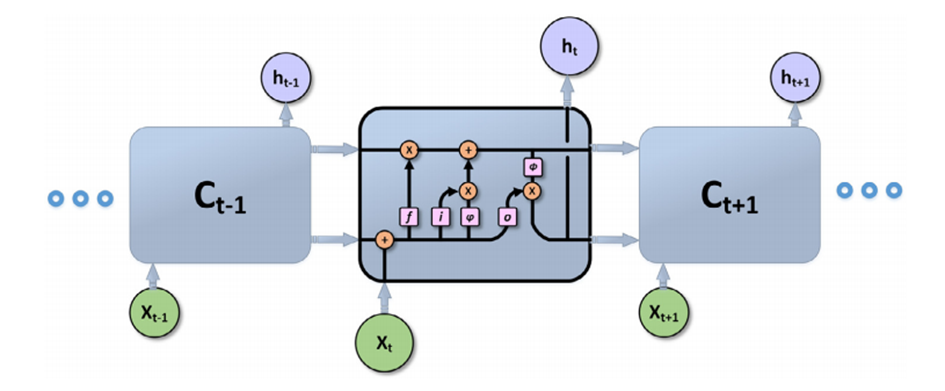


Figure 2.17 Vanilla LSTM architecture (Yuan, 2018)

In short, a Vanilla LSTM has the ability to add or delete information to the cell state. This can be regulated by the various gates present in LSTM namely; input, output and forget gate. The authors in (Dev, 2018) proposed a comparative study of a traditional LSTM and a deep neural network (DNN). The proposed models forecasted the daily and weekly movements of the Indian BSE sensex index. The traditional LSTM outperformed the DNN in weekly predictions. Another method (Ferdiansyah, 2019) focused on forecasting future bitcoin price using a standard LSTM model. The authors varied the number of epochs to attain the optimal LSTM model. In (Yong’an, 2020) the authors constructed a hybrid deep learning model for future stock price prediction using complementary ensemble empirical mode decomposition (CEEMD) as a sequence smoothing and decomposition module that can decompose the fluctuations of the time series data. The CEEMD module is followed by principal component analysis (PCA) that reduces the dimensionality of the decomposed components and eliminates the redundant information. The features are finally fed into the LSTM network to predict closing price for the next trading day.

In (David, 2017) the authors studied the usage of LSTM networks for the purpose of future stock price prediction and concluded that LSTM based models offer less risk and better accuracy as compared to the other strategies. In another approach (Baek, 2018), the authors proposed a novel data augmentation approach for future stock price prediction through the ModAugNet framework consisting of an overfitting prevention LSTM module and a prediction LSTM module. The proposed model was evaluated on S&P500 and Korea Composite Stock Price Index 200 (KOSPI200). The results confirmed the excellent forecasting accuracy of the proposed model.

The recent works (Mehtab, 2019) have opted for a hybrid approach for future stock price prediction by integrating deep learning and natural language processing. The hybrid model was evaluated on NIFTY 50 index values of NSE India for the period from January 2018 to June 2019 with a prediction horizon of one week. The LSTM model was augmented with a sentiment analysis module on Twitter data to correlate the public sentiments and the market sentiments. Similar to (Mehtab, 2019), the authors in (Cheng, 2018) proposed a hybrid approach for future stock price prediction. The authors proposed an attention-based LSTM model by adding an attention mechanism as an effective way to enhance the model’s capability and interpretability.

In (Cao, 2019) the authors proposed a hybrid approach for future stock price predictions involving Empirical Mode Decomposition (EMD) and LSTM model. The financial time series was decomposed into several intrinsic mode functions of different time scales using the EMD module which were then fed to LSTM prediction module for future stock price predictions. The proposed model outperformed Vanilla LSTM, SVM model, and multi-layer perceptron (MLP) model in predicting daily closing prices of S&P500, HSI, DAX, and SSE.

In another approach (Ghosh, 2020), the authors employed a hybrid model comprising of random forests and LSTM network for the purpose of forecasting out of sample directional movements of S&P 500 for the period from January 1993 to December 2018. The authors concluded that the hybrid model outperformed the traditional LSTM model.

**2.4.4.2 Bi-Directional LSTM**

In a Bi-Directional LSTM inputs flow in two directions; forward (past to future) and backwards (future to past), making a Bi-Directional LSTM different from a common LSTM. Bi-Directional LSTMs effectively increase the amount of information available to the network. With this form of network, the output layer can get information from past and future states simultaneously. The architecture of a Bi-Directional LSTM is illustrated in figure 2.18.

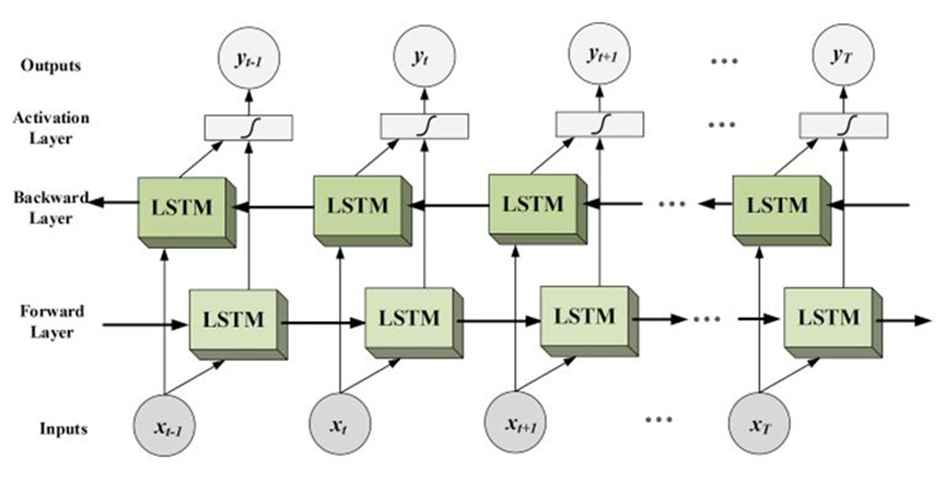


Figure 2.18 Bi-Directional LSTM Architecture

A Bi-Directional LSTM model has the ability to process and learn from the data in both directions i.e. from future to past and from past to future. Bi-Directional LSTM combines forward and backward contextual information and uses it to make predictions. In (Khaled, 2018) the authors compared and evaluated Bi-Directional LSTM and stacked LSTM models. The Bi-Directional LSTM demonstrated better performance and convergence for both short-term predictions and long-term predictions. The recent works (Eapen, 2019) have proposed a hybrid deep learning model that combines multiple pipelines of convolutional neural network (CNN) and Bi-Directional LSTM units. The authors observed that CNN layers show improved prediction performance when combined with Bi-Directional LSTM as compared with the traditional SVM regressor. Furthermore, the authors observed that multiple pipelines of deep layers perform better than single pipeline model. The architecture of the multiple pipeline model with CNNs followed by Bi-Directional LSTM is illustrated in figure 2.19.

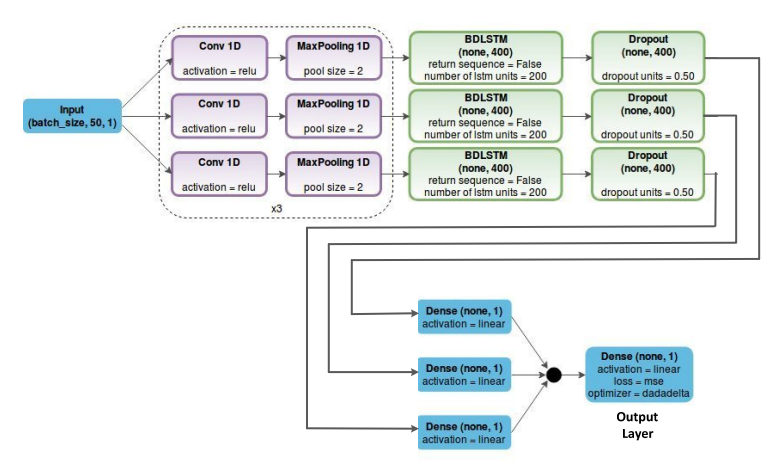


Figure 2.19 Architecture of the multiple pipeline model (Eapen, 2019)

**2.5 Discussion**

In the field of predictive modeling in future stock price prediction, various linear and nonlinear approaches have been put forward to predict future stock price including regression models, time series forecasting, support vector machines and neural networks. Recent studies in the field of stock price prediction may imply that neural networks can be a promising alternative to the traditional linear models because of their nonlinear modelling capability. This dominance can be attributed to the use of various neural networks such as ANNs, CNNs, RNNs, LSTM and their hybrid models. Among the various neural networks, RNNs are considered to be good at modeling and processing sequential data, which is quite suitable for stock price prediction. However, RNNs are not capable of learning long-term dependencies due to the popular vanishing gradient problem. This means that RNNs face difficulties in retaining for a long period of time. LSTMs are specifically designed to tackle the long-term dependency problem. Though the RNNs and LSTM models have obvious advantages in nonlinear data forecasting, their accuracy can still be improved. The authors in (Wang, 2020) proposed that the forecasting accuracy of traditional deep learning models alone is relatively low against the hybrid models. The authors in (Tai Wu, 2021) concluded that the neural network framework when combined with CNN and LSTM units showed better performance than the traditional CNN and LSTM models. The hybrid models have gained significant recognition through the studies illustrated in table 2.3.

Table 2.3 Studies following hybrid approach in Predictive Modeling

|  |  |  |
| --- | --- | --- |
| Survey Title | Year | Publication |
| Time series forecasting using a hybrid ARIMA and neural network model | 2003 | ELSEVIER |
| A graph‑based CNN‑LSTM stock price prediction algorithm with leading indicators | 2021 | SPRINGER |
| A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM | 2020 | ELSEVIER |
| Applied attention-based LSTM neural networks in stock prediction | 2018 | IEEE |
| Forecasting directional movements of stock prices for intraday trading using LSTM and random forests | 2021 | ELSEVIER |
| A CNN-LSTM-Based Model to Forecast Stock Prices | 2020 | Hindawi |
| Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction | 2019 | IEEE |
| Stock Price Prediction Using CNN and LSTM Based Deep Learning Models | 2020 | IEEE |
| ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module | 2018 | ELSEVIER |
| A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing | 2019 | IEEE |
| Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction | 2018 | IEEE |
| Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model | 2020 | IEEE |

**2.6 Summary**

This chapter provides a detailed review on predictive modeling in stock price prediction. This chapter reviews the linear and non-linear prediction frameworks including regression models, time series forecasting, support vector machines, neural networks and their hybrid models. This chapter highlights the advantages of using a hybrid approach over standalone traditional models. This chapter compares the Vanilla LSTM and Bi-Directional LSTM architecture and processing for model development and implementation.

**CHAPTER 3**

**RESEARCH METHODOLOGY**

**3.1 Introduction**

This chapter surrounds the research methodology and the supporting theoretical structure with detailed explanation of the steps involved. This chapter is dedicated to expound the technical jargons used throughout the study. This chapter begins with data selection followed by the data pre-processing techniques. The data pre-processing techniques include feature elimination, feature scaling, missing value treatment, exploratory data analysis and train-test split. This chapter highlights the benefits of data visualization and addresses different scaling techniques. This chapter concludes the advantages of the proposed deep learning models and their applications in stock price prediction. The performance metric section discusses the error metrics in detail.

**3.2 Research Approach**

The research approach follows some key processes involving data selection, data pre-processing, feature elimination, feature scaling, train-test split, model development and implementation and model evaluation. This approach focuses on analysing and developing a Vanilla LSTM and a Bi-Directional LSTM model and aims to predict future stock prices. The research approach is illustrated in the figure 3.1.

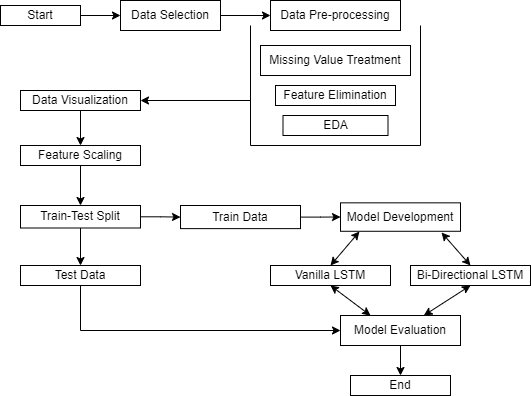


Figure 3.1 Research Approach

**3.2.1 Data Selection**

In stock price prediction, the selection of the financial time series data plays a prominent role as it determines the output of the prediction model. Large volume of stock market data is publicly available at various online websites. One such online website is Yahoo Finance which is part of the Yahoo network. Yahoo finance provides financial news, data and commentary including stock quotes, press releases, financial reports and some tools for personal finance management. Yahoo finance supports data extraction in various structured formats which may not require data cleaning. Generally, in stock price prediction the historical data contains the opening price, closing price, adjusted closing price, highest price and the lowest price of the stock for any particular day. The opening price and the closing price are prominent in predicting the future opening price and closing price of the stock.

**3.2.2 Data Pre-processing**

Data pre-processing is the first and crucial step while developing a machine learning model. Data pre-processing refers to the process of preparing and organizing the raw data to make it suitable for training a machine learning model. In data pre-processing it is essential to eliminate any missing value, noise and other anomalies in the raw data as any inconsistency in the raw data may lead to unreliable results. The data pre-processing techniques include missing value treatment, feature elimination, feature scaling, exploratory data analysis and train-test split.

**3.2.2.1 Missing Value Treatment**

In data pre-processing it is essential to eliminate any missing value, noise and other anomalies in the raw data as any inconsistency in the raw data may lead to unreliable results. The real world data or the raw data may or may not have missing values present within. The cause behind missing values can be data corruption or data unavailability. Missing value treatment is a very crucial task in data pre-processing as many machine learning algorithms do not support missing values or the missing values may bias the model output. The missing value treatment involves either deleting the missing values or imputing the missing values. The missing values can be imputed with mean, median and mode or with an arbitrary value that fits the domain. The missing values can be imputed via sci-kit learn library as well. The missing values in the selected dataset are to be detected and treated accordingly.

**3.2.2.2 Feature Elimination**

Various machine learning algorithms are prone to curse of dimensionality. The curse of dimensionality refers to the exponential amount of computational efforts required for analysing and processing high dimensional data. Feature elimination gives an effective way to overcome challenges like curse of dimensionality, overfitting, learning accuracy and computational time. In data pre-processing it is crucial to identify the important features or variables that are required to achieve the desired results. Feature elimination can either be performed manually or via an algorithm. One such algorithm is Recursive Feature Elimination (RFE). RFE is a feature selection method that fits a model and removes the weakest features until the specified number of features is reached. The important features required for the purpose of the study are to be ensured via feature elimination.

**3.2.2.3 Exploratory Data Analysis**

Exploratory Data Analysis (EDA) is an approach to analyse the data using visual techniques. It is used to discover trends and patterns or to detect outliers with the help of statistical summary and graphical representations. EDA takes around 40-50% of total project time as the model outcome depends on the quality of input data being fed to train the model. EDA involves univariate analysis and bivariate analysis.

Univariate analysis is the simplest form of analysing data. Univariate analysis requires to analyse each feature separately. It can be either inferential or descriptive. A descriptive Univariate analysis provides the descriptive statistics of each feature that can help in detecting outliers in the data. An outlier is an observation that is numerically distant from the rest of the data. Boxplots are particularly used to detect outliers within a dataset. A boxplot gives a good indication of the data distribution. A boxplot can graphically demonstrate the locality, spread and skewness of the numerical data through their quartiles. A vertical line goes through the box at the median.

Bivariate analysis is a statistical analysis where two variables are analysed for the purpose of determining the empirical relationship between them. For the scope of future stock price prediction, bivariate analysis is not required.

**3.2.2.4 Data Visualization**

Data visualization techniques are progressing exponentially in the field of machine learning and data analysis. These techniques enhance the understanding of the data and provide a first-hand overview into the pattern followed by the data. Data visualization techniques focus on attaining a better comprehension of data in graphical formats such as charts, graphs and maps. Data visualization techniques provide an accessible way to see and detect trends, outliers and patterns in data. Some of the most common data visualization libraries are Matplotlib and Seaborn.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. This library is built on numpy arrays and consists of several plots like line, bar, scatter and histogram.

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It provides beautiful default styles and colour palettes to make statistical plots more attractive. Seaborn aims to make visualization the central part of exploring and understanding data.

**3.2.2.5 Feature Scaling**

A machine learning algorithm tends to weigh greater values as higher and smaller values as lower, regardless of the unit of the values. Scaling techniques are used to handle highly varying magnitudes or values of the features. Scaling techniques are often used prior to the train-test split procedure. The two common techniques to perform feature scaling are Min-Max Normalization and Standardization.

Min-Max Normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets converted into zero, the maximum value gets converted into one and the other values get converted into a decimal number between zero and one. The expression for Min-Max scaling is as follows.

Standardization is the technique of scaling the features such that their mean gets converted to zero and standard deviation gets converted to one. In Standardization the values get centred on the mean with a unit standard deviation. The expression for Standardization is as follows.

**3.2.2.6 Train-Test Split**

Train-Test split is a technique for evaluating the performance of a machine learning algorithm. It is a fast and easy technique to perform and can be used for any supervised learning algorithm. Although it is simple to use and interpret, there are times when the technique should not be used such as situations involving a small dataset or class imbalance. The train-test split procedure is appropriate in case of large datasets. The procedure involves splitting the dataset into two subsets. The first subset is used to fit or train the model and is referred to as the training dataset. The second subset is fed to the model as input and is referred to as test dataset.

**3.2.3 Model Development**

The machine learning model development cycle involves data acquisition, data pre-processing to make the data suitable for building the model, choosing the appropriate algorithm to build the model, training the model and hyper parameter tuning to select the best performing model. Neural networks have number of hidden layers, batch size, number of epochs and number of time steps as the hyper parameters which can be intuitively tuned for improving model accuracy. An LSTM model can only accept a three-dimensional array as an input where the first dimension represents the batch size, the second dimension represents the number of time steps and the third dimension represents the number of units in one input sequence. The batch size refers to the number of training samples utilized in one iteration. A time step is a single occurrence of a cell. An epoch means one complete pass of the train dataset through the model. This section discusses the model development process of two LSTM models i.e. Vanilla LSTM and Bi-Directional LSTM. The sequential model can be imported from Keras or other required libraries in Google Colab. Colab notebooks allow users to combine executable code and rich text in a single document, along with images.

**3.2.3.1 Vanilla LSTM**

A Vanilla LSTM is an LSTM model that has a single hidden layer of LSTM units and an output layer used to make predictions. The train data and the test data are to be converted into a three-dimensional sequence. The hyper parameters involved in building a Vanilla LSTM are LSTM units, drop out percentage, time steps, number of epochs, lost function and the performance metric. The input sequence is to be fed to the model for the training process via model.fit() function. After the training process, the model is to be validated or evaluated on the test data.

**3.2.3.2 Bi-Directional LSTM**

In a Bi-Directional LSTM inputs flow in two directions; forward (past to future) and backwards (future to past), making a Bi-Directional LSTM different from a Vanilla LSTM. The train data and the test data are to be converted into a three-dimensional sequence. The hyper parameters involved in building a Bi-Directional LSTM are number of dense layers, LSTM units, drop out percentage, time steps, number of epochs, lost function and the performance metric. The input sequence is to be fed to the model for the training process via model.fit() function. After the training process, the model is to be validated or evaluated on the test data.

**3.2.4 Model Evaluation**

Model evaluation is the process of using different performance metrics to understand the machine learning model's performance, strengths and weaknesses. Model evaluation is important to assess the efficacy of the model. The performance metrics vary as per the machine learning model. For a classification model, the performance metrics can be accuracy, precision and recall. For a regression model, the performance metrics can be RMSE or MAPE.

Root Mean Square Error is a frequently used performance metric that evaluates the difference between the actual values and the values predicted by the machine learning model.

Mean Absolute Percentage Error (MAPE) is one of the most commonly used performance metric to measure forecast accuracy. It is the mean or average of the absolute percentage errors of the forecasts.

**3.3 Summary**

This chapter provides an overview of the research approach followed and the supporting theoretical structure with detailed explanation of the steps involved. This chapter illustrates the research approach followed via a flow chart. This chapter discusses the selection of financial time series data and the required data pre-processing steps. This chapter provides insights on the processes involved in model development and model evaluation. This chapter highlights two commonly used data visualization libraries i.e. Matplotlib and Seaborn. This chapter explains two performance metrics i.e. RMSE and MAPE.

**CHAPTER 4**

**ANALYSIS AND DESIGN**

**4.1 Introduction**

This chapter comprises of the data pre-processing and model design whereby the sub-chapters include the data preparation, exploratory data analysis, data visualization, model development and model implementation. This chapter begins with the detailed description of the dataset used in the study followed by the data pre-processing techniques involved for further analysis. These techniques include feature elimination, feature engineering, missing value treatment, univariate analysis and train-test split. Univariate analysis determines the distribution of the features present in the dataset. The features are visualized via two popular plotting libraries i.e. Matplotlib and Seaborn.

**4.2 Dataset Description**

The dataset used in the study is publicly available and is obtained from Yahoo Finance. The dataset comprises of historical stock data i.e. Open, High, Low, Close, Adj Close and Volume features of three NSE stocks i.e. HDFC, Reliance and Tata Motors for the period of six months, starting from 9 August 2021 to 8 February 2022. The dataset has time span of six months and contains every day’s opening value, high value, low value, closing value and adjusted closing value which can be used to predict the future opening value and closing value of the stocks.

**4.3 Dataset Preparation**

The raw data is obtained from Yahoo Finance in a csv file. Although the raw data obtained is in structured format, it is in best practice to pre-process the raw data before the train-test split. Some of the data pre-processing techniques involve feature elimination, feature engineering, missing value treatment and univariate and bivariate analysis.

**4.3.1 Missing Value Treatment**

The real world data or the raw data may or may not have missing values present in it. The cause behind missing values can be data corruption or data unavailability. Missing value treatment is a very crucial task in data pre-processing as many machine learning algorithms do not support missing values or the missing values may bias the model output.

The dataset comprises of Open, High, Low, Close, Adj Close and Volume features. These features range from 9 August 2021 to 8 February 2022 and do not have any missing values present within them. Thus, the missing value treatment is not required.

**4.3.2 Feature Elimination**

In the stage of dataset preparation or data pre-processing, it is crucial to identify the important features or variables that are required to achieve the desired results as the unrelated variables are not required to meet the goal of the study. One of the steps of data pre-processing is to eliminate unnecessary features as these features are not required to meet the aim and objectives of the study.

The dataset comprises of Open, High, Low, Close, Adj Close and Volume features. In order to predict the future opening value or closing value of stocks, High, Low, Adj close and Volume features are not required. Thus, these features are removed from the original dataset.

**4.3.3 Univariate Analysis**

To further understand the distribution of values in each feature univariate analysis is performed. Univariate analysis requires each feature to be analysed separately. Univariate analysis provides the descriptive statistics of each feature that help in detecting outliers in the data. A boxplot gives a good indication of the data distribution. A boxplot can graphically demonstrate the locality, spread and skewness of the numerical data through their quartiles. A vertical line goes through the box at the median.

Boxplots are particularly used to detect outliers within a dataset. An outlier is an observation that is numerically distant from the rest of the data. When reviewing a boxplot, an outlier is identified as a data point that is located outside the whiskers of the box plot. The boxplots for Open and Close Price of Tata Motors, HDFC and Reliance are illustrated in the figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 respectively.



Figure 4.1 Boxplot of Open Price of Tata Motors



Figure 4.2 Boxplot of Close Price of Tata Motors

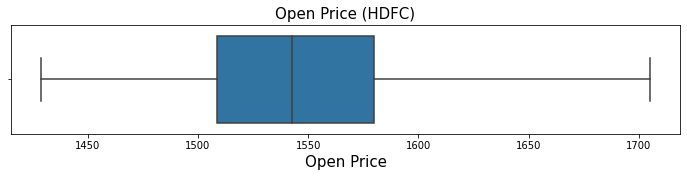


Figure 4.3 Boxplot of Open Price of HDFC

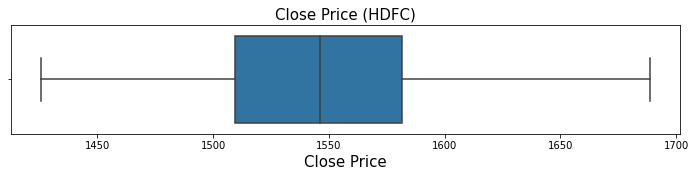


Figure 4.4 Boxplot of Close Price of HDFC

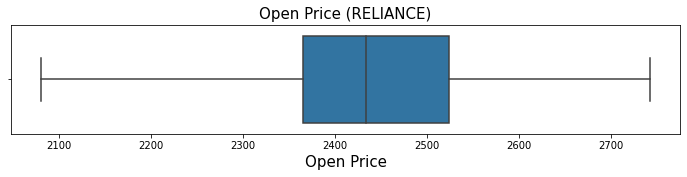


Figure 4.5 Boxplot of Open Price of Reliance

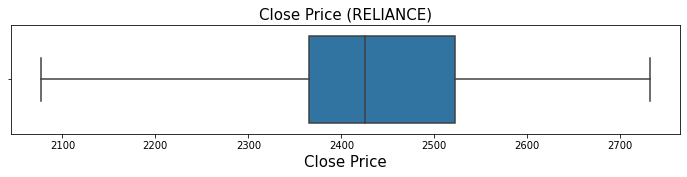


Figure 4.6 Boxplot of Close Price of Reliance

There are no data points present outside the whiskers of the boxplots. There are no outliers present in the dataset. Thus, the outlier treatment is not required.

**4.3.4 Data Visualization**

Data visualization is the graphical representation of information and data. The visual elements used in data visualization can be charts, maps and graphs. Data visualization elements provide an accessible way to see and detect trends, outliers and patterns in data.

The visual representation of the Open and Close Price of Tata Motors, HDFC and Reliance is illustrated in the figures 4.7, 4.8 and 4.9 respectively.

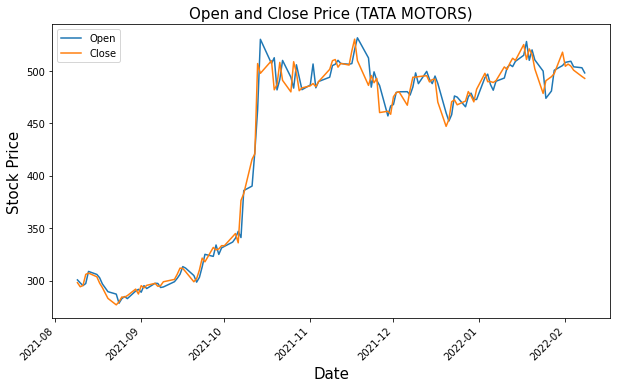


Figure 4.7 Open and Close Price of Tata Motors

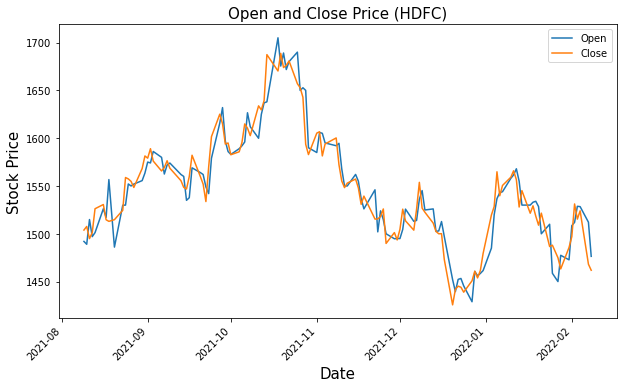


Figure 4.8 Open and Close Price of HDFC

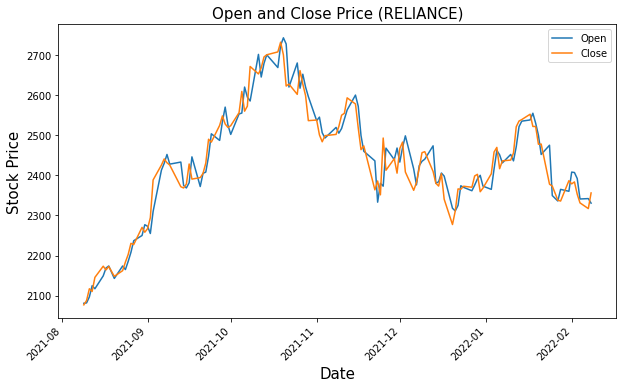


Figure 4.9 Open and Close Price of Reliance

**4.3.5 Feature Engineering**

Feature engineering is the process of selecting, manipulating and transforming raw data into features that can be used in machine learning algorithms. It is the process of converting raw observations into desired features using statistical or machine learning approaches. The goal of feature engineering is to enhance the model accuracy.

The dataset comprises of Open and Close features which refer to the opening price and closing price of the stocks. These features are sufficient to predict the future opening price and closing price of stocks. Thus, feature engineering is not required.

**4.3.6 Feature Scaling**

A machine learning algorithm tends to weigh greater values as higher and smaller values as lower, regardless of the unit of the values. Scaling techniques are used to handle highly varying magnitudes or values of the features. Scaling techniques are often used prior to the train-test split procedure. The two common techniques to perform feature scaling are Min-Max Normalization and Standardization.

The Open and Close Price of HDFC, Reliance and Tata Motors are scaled using the Min-Max Normalization scaling technique. This technique scales the features into the range of zero and one using the following expression.

**4.3.7 Train-Test Split**

Train-Test split is a technique for evaluating the performance of a machine learning algorithm. It is a fast and easy technique to perform and can be used for any supervised learning algorithm. Although it is simple to use and interpret, there are times when the technique should not be used such as situations involving a small dataset or class imbalance. The train-test split procedure is appropriate in case of large datasets. The procedure involves splitting the dataset into two subsets. The first subset is used to fit or train the model and is referred to as the training dataset. The second subset is fed to the model as input and is referred to as test dataset.

The dataset comprises of Open and Close features with a total of 126 scaled records. The dataset is split into train data and test data with a ratio of 7:3. The train dataset comprises of Open and Close features with a total of 88 records and the test dataset comprises of Open and Close features with a total of 38 records.

**4.4 Model Development and Implementation**

The sequential model is imported from Keras and other required libraries. An LSTM model can only accept a three-dimensional array as an input. The first dimension represents the batch size, the second dimension represents the time-steps and the third dimension represents the number of units in one input sequence. The batch size refers to the number of training samples utilized in one iteration. A time step is a single occurrence of a cell. An epoch means one complete pass of the train dataset through the model. The batch size, time-steps, epochs, and dense layers are some of the hyper parameters involved in model development.

**4.4.1 Vanilla LSTM**

A Vanilla LSTM is an LSTM model that has a single hidden layer of LSTM units, and an output layer used to make a prediction. The train dataset has a total of 88 records and the test dataset has a total of 38 records. The train data and test data are converted into a three-dimensional sequence where the first dimension represents the batch size, the second dimension represents the time-steps and the third dimension represents the number of units in one input sequence.

The number of dense layers for a Vanilla LSTM is one. The number of units assigned is 50 with a drop out of 10%. The number of time-steps assigned is 10 and the number of epochs assigned is 80. The lost function used is mean squared error (MSE) and the error metric used is mean absolute percentage error (MAPE). The model is trained on the train data and validated on the test data for HDFC, Reliance and Tata Motors. The execution time and MAPE for Tata Motors, HDFC and Reliance is illustrated in table 4.1.

Table 4.1 Execution time and MAPE for respective stocks (Vanilla LSTM)

|  |  |  |  |
| --- | --- | --- | --- |
| Stock | Execution time | MAPE (Open Price) | MAPE (Close Price) |
| TATA MOTORS | 11s | 2.67 | 2.19 |
| HDFC | 13s | 2.44 | 2.59 |
| RELIANCE | 9s | 2.94 | 2.82 |

The visual representation of the actual Open Price and the predicted Open Price for Tata Motors, HDFC and Reliance is illustrated in figure 4.10, 4.11 and 4.12 respectively.

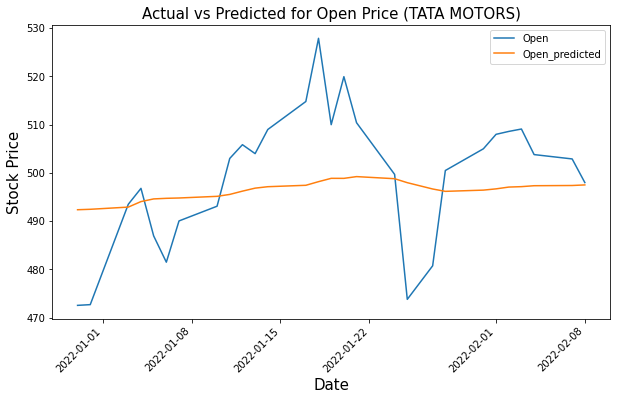


Figure 4.10 Actual v/s Predicted for Open Price of Tata Motors (Vanilla LSTM)

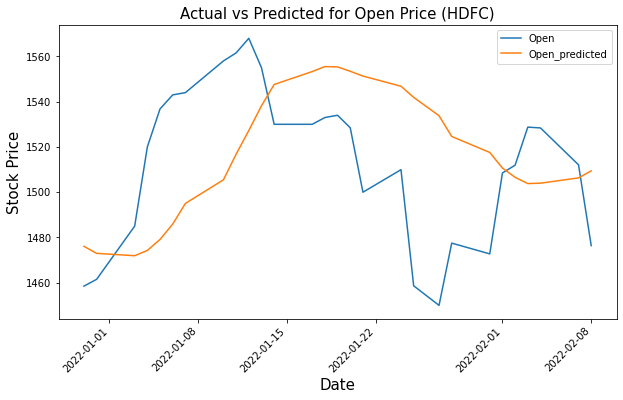
****

Figure 4.11 Actual v/s Predicted for Open Price of HDFC (Vanilla LSTM)

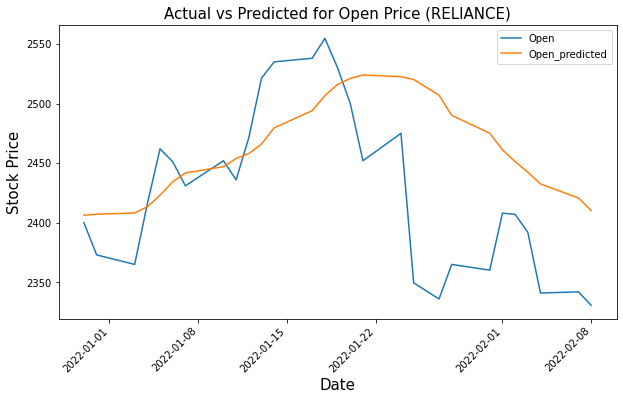


Figure 4.12 Actual v/s Predicted for Open Price of Reliance (Vanilla LSTM)

The visual representation of the actual Close Price and the predicted Close Price for Tata Motors, HDFC and Reliance is illustrated in figure 4.13, 4.14 and 4.15 respectively.

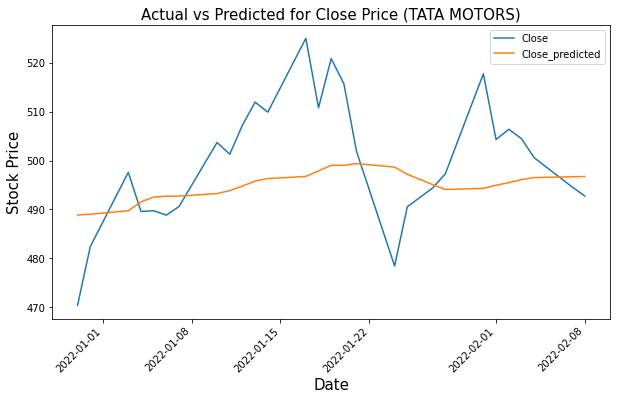


Figure 4.13 Actual v/s Predicted for Close Price of Tata Motors (Vanilla LSTM)

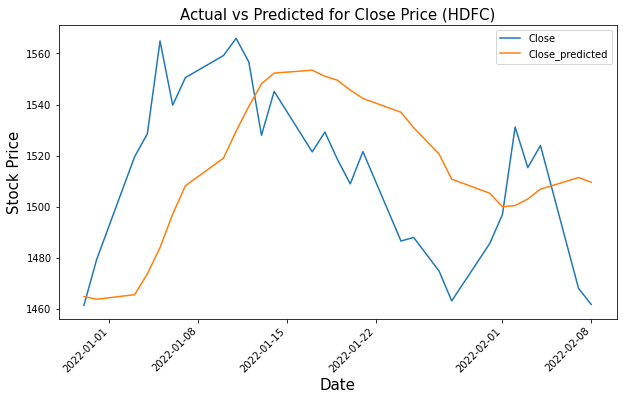


Figure 4.14 Actual v/s Predicted for Close Price of HDFC (Vanilla LSTM)

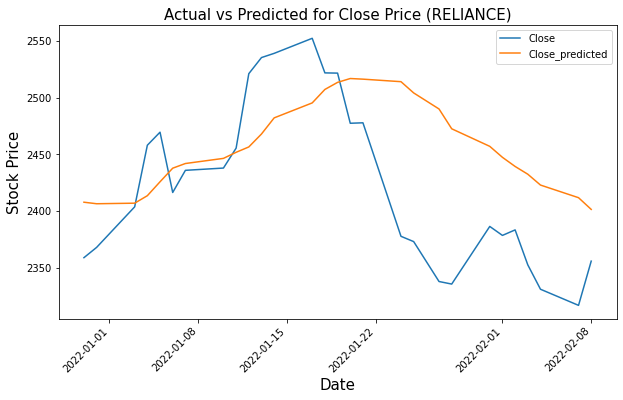


Figure 4.15 Actual v/s Predicted for Close Price of Reliance(Vanilla LSTM)

**4.4.2 Bi-Directional LSTM**

In a Bi-Directional LSTM inputs flow in two directions; forward (past to future) and backwards (future to past), making a Bi-Directional LSTM different from a vanilla LSTM. The number of dense layers assigned is two. The number of units assigned is 50 with a drop out of 10%. The number of time-steps assigned is 10 and the number of epochs assigned is 80. The lost function used is mean squared error (MSE) and the error metric used is mean absolute percentage error (MAPE). The model is trained on the train data and validated on the test data for HDFC, Reliance and Tata Motors. The execution time and MAPE for Tata Motors, HDFC and Reliance is illustrated in table 4.2.

Table 4.2 Execution time and MAPE for respective stocks (Bi-LSTM)

|  |  |  |  |
| --- | --- | --- | --- |
| Stock | Execution time | MAPE (Open Price) | MAPE (Close Price) |
| TATA MOTORS | 16s | 1.75 | 2.0 |
| HDFC | 19s | 1.14 | 1.55 |
| RELIANCE | 17s | 1.64 | 1.68 |

The visual representation of the actual Open Price and the predicted Open Price for Tata Motors, HDFC and Reliance is illustrated in figure 4.16, 4.17 and 4.18 respectively.

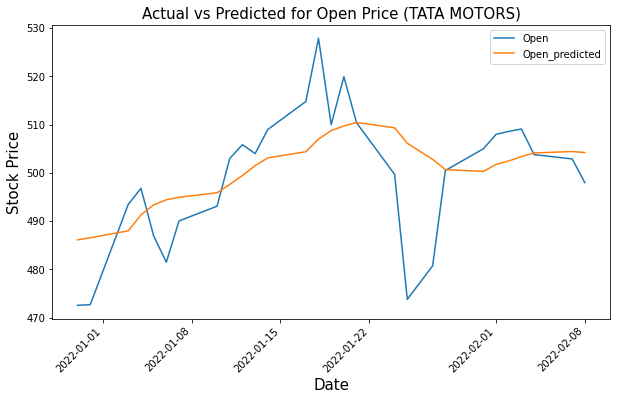
****

Figure 4.16 Actual v/s Predicted for Open Price of Tata Motors (Bi-Directional LSTM)

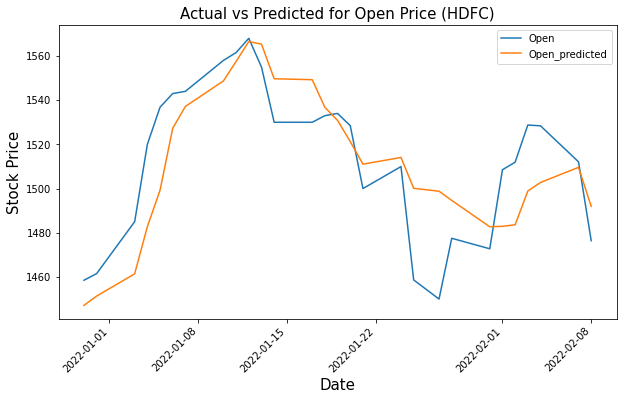
****

Figure 4.17 Actual v/s Predicted for Open Price of HDFC (Bi-Directional LSTM)

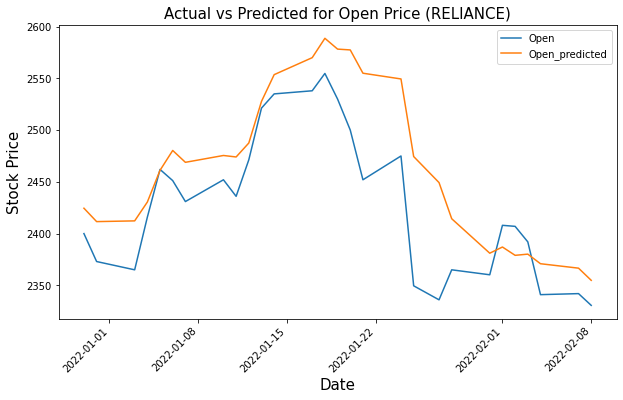
****

Figure 4.18 Actual v/s Predicted for Open Price of Reliance (Bi-Directional LSTM)

The visual representation of the actual Close Price and the predicted Close Price for Tata Motors, HDFC and Reliance is illustrated in figure 4.19, 4.20 and 4.21 respectively.

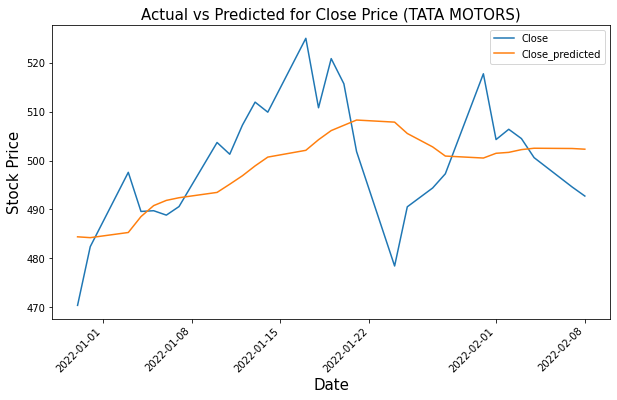


Figure 4.19 Actual v/s Predicted for Close Price of Tata Motors (Bi-Directional LSTM)

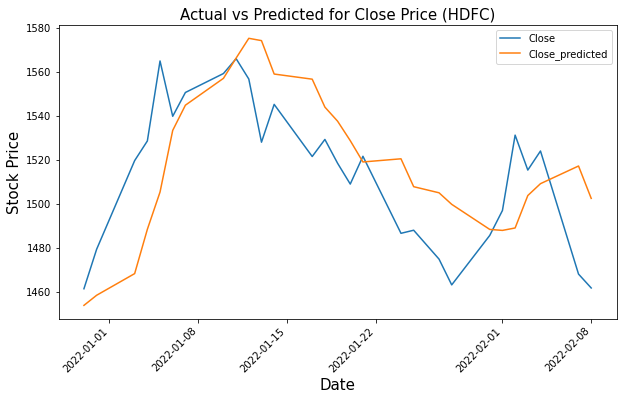
****

Figure 4.20 Actual v/s Predicted for Close Price of HDFC (Bi-Directional LSTM)

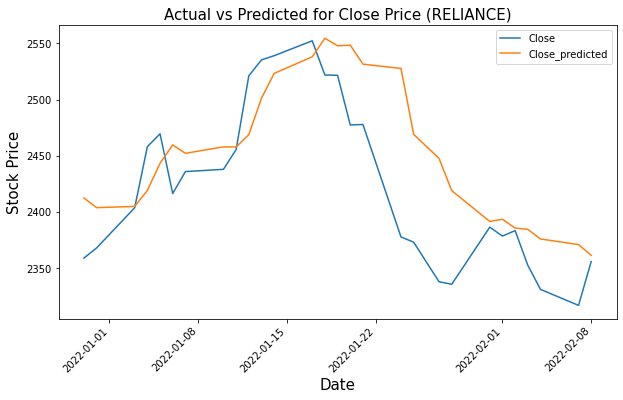
****

Figure 4.21 Actual v/s Predicted for Close Price of Reliance (Bi-Directional LSTM)

**4.5 Summary**

This chapter provides a detailed explanation on data pre-processing and model development and implementation. The raw data obtained from Yahoo Finance is pre-processed and visualized before the train-test split. The train data and test data is converted into a three dimensional sequence where the first dimension represents the batch size, the second dimension represents the time-steps and the third dimension represents the number of units in one input sequence. The sequential model is imported from Keras and the required libraries. The Vanilla LSTM is trained on the train data with a single dense layer. The Bi-Directional LSTM is trained on the train data with two dense layers present. The number of time steps assigned is 10 and the number of epochs assigned is 80 for both the models. Although the execution time for the Bi-Directional LSTM is high, the Bi-Directional LSTM has yielded better results with low MAPE values as compared to the Vanilla LSTM.

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

**5.1 Introduction**

This chapter includes all the interpretations from the analysis and the results of the study whereby the sub-chapters include the interpretations of visualizations, model evaluation, visualizations for the optimal model, results and discussions. This chapter begins with detailed interpretation of the visualizations followed by the model evaluation. The factors used for model evaluation is the performance metric MAPE and the execution time. The results are compared and discussed with the state of the art prediction frameworks.

**5.2 Interpretations of Visualizations**

The various visualizations exhibited in chapter 4.4 display the actual stock trend v/s the predicted stock trend for the period from 30 December 2021 to 08 February 2022. The trend lines in the visualizations exhibited by the Vanilla LSTM for the Open Price/Close Price and predicted Open Price/Close Price of Tata Motors, HDFC and RELIANCE are quite distant as compared to the trend lines in the visualizations exhibited by the Bi-Directional LSTM for the Open Price/Close Price and predicted Open Price/Close Price of Tata Motors, HDFC and RELIANCE despite having the same hyper parameters. The Bi-Directional LSTM has outperformed the Vanilla LSTM in predicting the future stock prices.

**5.3 Model Evaluation**

The sequential model is imported from Keras and the required libraries. The batch size, time-steps, epochs, and dense layers are some of the hyper parameters involved in Vanilla LSTM and Bi-Directional LSTM models. These hyper parameters can be varied to control the learning process which leads to various different models. The models are assessed on factors including execution time and MAPE.

**5.3.1 Vanilla LSTM**

A Vanilla LSTM is an LSTM model that has a single hidden layer of LSTM units, and an output layer used to make a prediction. The various Vanilla LSTM models implemented are illustrated in the table 5.1 and table 5.2.

Table 5.1 Various Vanilla LSTM models for Open Price

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TATA MOTORS | | HDFC | | RELIANCE | |
| Number of Epochs | Execution time | MAPE | Execution time | MAPE | Execution time | MAPE |
| 50 | 7s | 2.83 | 9s | 2.71 | 11s | 3.11 |
| 80 | 11s | 2.67 | 13s | 2.44 | 9s | 2.94 |
| 120 | 12s | 2.51 | 20s | 2.41 | 15s | 2.67 |
| 150 | 14s | 2.69 | 25s | 2.44 | 17s | 2.71 |

Table 5.2 Various Vanilla LSTM models for Close Price

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TATA MOTORS | | HDFC | | RELIANCE | |
| Number of Epochs | Execution time | MAPE | Execution time | MAPE | Execution time | MAPE |
| 50 | 7s | 2.75 | 9s | 2.78 | 11s | 3.13 |
| 80 | 11s | 2.19 | 13s | 2.59 | 9s | 2.82 |
| 120 | 12s | 2.17 | 20s | 2.53 | 15s | 2.73 |
| 150 | 14s | 2.41 | 25s | 2.61 | 17s | 2.63 |

It is observed that with an increase in the number of epochs the execution time also increases, however with an increase in the number of epochs the mean absolute percentage error (MAPE) decreases up to an extent. The MAPE values decreased up till 120 epochs. The MAPE values for 150 epochs are either the same or slightly higher than that for 120 epochs. Thus, the optimal Vanilla LSTM model has 120 number of epochs with a dropout percentage of 10. The time steps assigned is 10 and the number of LSTM units assigned is 50.

**5.3.2 Bi-Directional LSTM**

In a Bi-Directional LSTM inputs flow in two directions; forward (past to future) and backwards (future to past), making a Bi-Directional LSTM different from a vanilla LSTM. The various Bi-Directional LSTM models implemented are illustrated in the table 5.3 and table 5.4.

Table 5.3 Various Bi-Directional LSTM models for Open Price

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TATA MOTORS | | HDFC | | RELIANCE | |
| Number of Epochs | Execution time | MAPE | Execution time | MAPE | Execution time | MAPE |
| 50 | 16s | 1.75 | 17s | 1.71 | 11s | 1.88 |
| 80 | 16s | 1.62 | 19s | 1.14 | 17s | 1.64 |
| 120 | 21s | 1.46 | 20s | 0.97 | 23s | 0.8 |
| 150 | 23s | 1.81 | 27s | 0.97 | 30s | 0.94 |

Table 5.4 Various Bi-Directional LSTM models for Close Price

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TATA MOTORS | | HDFC | | RELIANCE | |
| Number of Epochs | Execution time | MAPE | Execution time | MAPE | Execution time | MAPE |
| 50 | 16s | 2.0 | 17s | 1.75 | 11s | 2.44 |
| 80 | 16s | 1.76 | 19s | 1.55 | 17s | 1.68 |
| 120 | 21s | 1.71 | 20s | 1.31 | 23s | 1.3 |
| 150 | 23s | 1.73 | 27s | 1.38 | 30s | 1.47 |

It is observed that the execution time for the various Bi-Directional models is higher than that for the Vanilla LSTM models whereas the MAPE value for the Bi-Directional models is significantly lower than that for the Vanilla LSTM models despite having the same hyper parameters. Although the execution time for the Bi-Directional LSTM model is higher than that for the Vanilla LSTM, it is evident that the Bi-Directional LSTM model has outperformed the Vanilla LSTM model. The MAPE values for the Bi-Directional model decreased up till 120 epochs like the Vanilla LSTM model. The MAPE values for 150 epochs are either the same or slightly higher than that for 120 epochs. Thus, the optimal Bi-Directional LSTM model has two dense layers, 120 number of epochs and a dropout percentage of 10. The time steps assigned is 10 and the number of LSTM units assigned is 50.

**5.4 Visualizations for the Optimal Model**

The optimal model for predicting the future Open Price and Close Price of Tata Motors, HDFC, Reliance is the Bi-Directional model with two dense layers, a dropout percentage of 10, 50 LSTM units, 10 time steps and 120 epochs. The visualizations for the optimal model are illustrated in figures 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6.

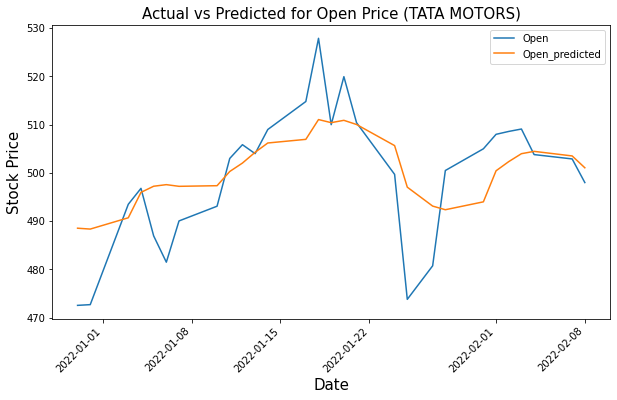


Figure 5.1 Actual v/s Predicted for Open Price of Tata Motors (Optimal Model)

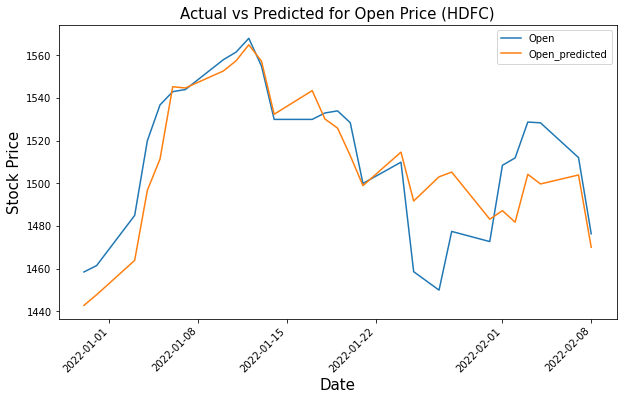


Figure 5.2 Actual v/s Predicted for Open Price of HDFC (Optimal Model)

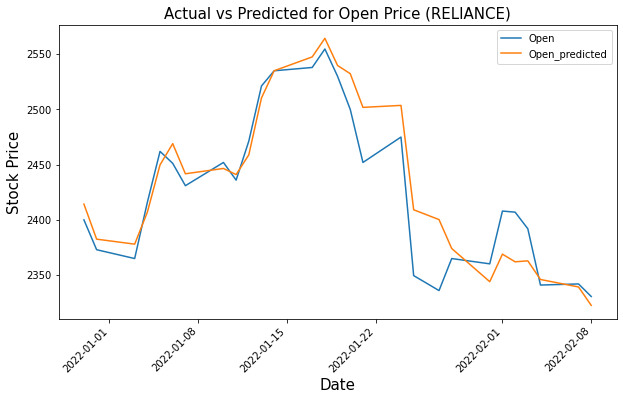


Figure 5.3 Actual v/s Predicted for Open Price of Reliance (Optimal Model)

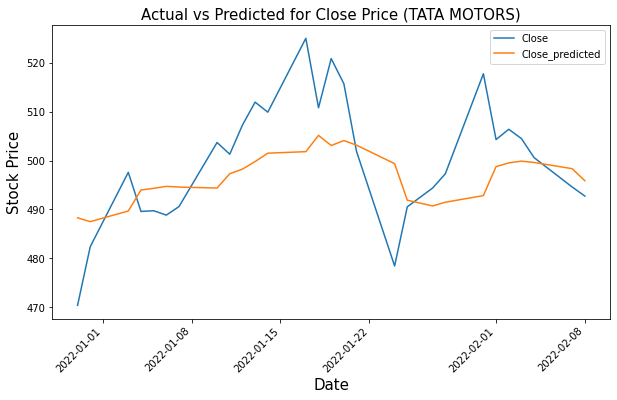


Figure 5.4 Actual v/s Predicted for Close Price of Tata Motors (Optimal Model)

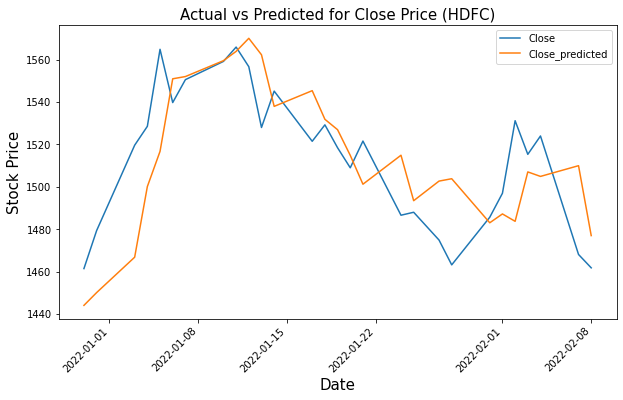


Figure 5.5 Actual v/s Predicted for Close Price of HDFC (Optimal Model)

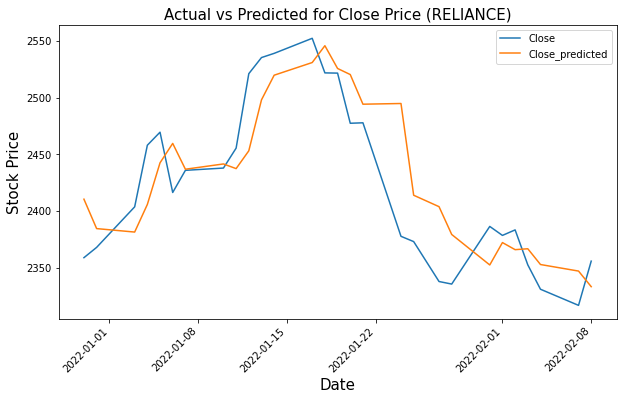


Figure 5.6 Actual v/s Predicted for Close Price of Reliance (Optimal Model)

**5.5 Discussions**

After proper scaling, training and testing of the Vanilla LSTM and Bi-Directional LSTM between the actual data and the predicted data, the various models are evaluated on the basis of their MAPE values. The MAPE values varied with the number of epochs, dense layers, time steps and the batch size. The Bi-Directional LSTM with 120 epochs, two dense layers, 50 LSTM units. 10 time steps and a dropout percentage of 10 has the lowest MAPE value and gives better prediction accuracy than the other models. Upon increasing the number of epochs from 120, to say 150, the model suffered from an overfitting issue. Although the Vanilla LSTM took less amount of time to forecast future stock price as compared to the Bi-Directional LSTM, the Bi-Directional LSTM has outperformed the Vanilla LSTM in future stock price prediction. The forecasting accuracy of the Bi-Directional LSTM is comparable with the state of the art predictions.

**5.6 Summary**

This chapter interprets the analysis and the results of the study. This chapter compares the visualizations of the Vanilla LSTM model and Bi-Directional LSTM model. This chapter evaluates the Vanilla LSTM and Bi-Directional LSTM models on the basis of their MAPE values. The Bi-Directional LSTM models have lower MAPE values as compared to the Vanilla LSTM models. The Bi-Directional LSTM with 120 epochs, two dense layers, 50 LSTM units. 10 time steps and a dropout percentage of 10 has the lowest MAPE value and is concluded as the optimal model. This chapter compares and discusses the results with the state of the art predictions.

**CHAPTER 6**

**CONCLUSIONS AND RECOMMENDATIONS**

**6.1 Introduction**

This chapter discusses the study in brief and draws conclusions based on the results. The discussion highlights the steps carried out for future stock price predictions via Vanilla LSTM and Bi-Directional LSTM. This chapter elucidates the results obtained from various analysis and model evaluation. This chapter addresses the aim and objectives of the study and their achievement. This chapter highlights the future recommendations and extensions of the study.

**6.2 Discussions and Conclusions**

This study has reviewed state of the art prediction frameworks used for future stock price prediction. The study is conducted using a publicly available dataset obtained from Yahoo Finance. The dataset comprises of historical stock data i.e. Open, High, Low, Close, Adj Close and Volume features of three NSE stocks i.e. HDFC, Reliance and Tata Motors for the period of six months, starting from 9 August 2021 to 8 February 2022. The dataset is pre-processed with feature elimination and feature scaling before the train-test split. The train data and test data is converted into a three-dimensional sequence. The models are trained on the train data and validated on the test data. The models are compared by varying the number of epochs, hidden layers, batch size, LSTM units and time steps. The models are evaluated on the basis of their MAPE values.

This study concludes that combination of statistical models and deep learning models such as ANNs, RNNs, and LSTMs can reveal complex non-linear patterns of financial time series data that are difficult to detect with linear algorithms. Deep learning models have proved to be more effective in future stock price prediction than the linear regression models. The reason for their dominance in time series application is that deep learning models are commonly utilized to perform tasks involving sequential input data such as financial time series. However, recent studies have revealed that hybrid approaches comprising of various deep learning models can outperform standalone deep learning models. This study concludes that a Bi-Directional LSTM outperforms a Vanilla LSTM in future stock price prediction.

**6.3 Contribution to Knowledge**

This study contributes to the field of predictive modeling in future stock price prediction. This study concludes that a Bi-Directional LSTM outperforms a Vanilla LSTM in future stock price prediction. This study can be used in future literature review involving Vanilla LSTM and Bi-Directional LSTM.

**6.4 Future Recommendations**

The future work can focus on introducing a hybrid approach by integrating Bi-Directional LSTM with natural language processing and incorporating sentiment analysis by involving public sentiments, policies and news surrounding the stocks from legitimate blogs, Twitter and Facebook. The forecasting accuracy may increase by introducing attention mechanism in the LSTM model and utilizing a relatively larger dataset.

**RERFERENCES**

1. Jingyang Wang (2020), “A CNN-LSTM-Based Model to Forecast Stock Prices” Volume 2020, Article ID 6622927, 10 pages
2. Jimmy Ming‑Tai Wu (2021), “A graph‑based CNN‑LSTM stock price prediction algorithm with leading indicators” Received: 14 October 2020 / Accepted: 30 January 2021
3. Gan Kim Soon (2013), “A Review of Stock Market Prediction with Artificial Neural Network (ANN)” 2013 IEEE International Conference on Control System, Computing and Engineering
4. Francis E.H. Tay (2001), “Application of support vector machines in financial time series forecasting” Department of Mechanical Engineering, National University of Singapore
5. Qihang Ma (2020), “Comparison of ARIMA, ANN and LSTM for Stock Price Prediction” E3S Web of Conferences 218, 01026 (2020)
6. Mirza Mohd Shahriar Maswood (2020), “Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model” 2nd Novel Intelligent and Leading Emerging Sciences Conference
7. Halbert White (2002), “ECONOMIC PREDICTION USING NEURAL NETWORKS: THE CASE OF IBM DAILY STOCK RETURNS” Department of Economics University of California, San Diego
8. Kyoung-jae Kim (2003), “Financial time series forecasting using support vector machines” Department of Information Systems, College of Business Administration
9. Jinquan Zhang (2018), “Forecasting of Forex Time Series Data Based on Deep Learning” 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018
10. Jithin Eapen (2019), “Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction”
11. Vijay Krishna Menon (2018), “NSE Stock Market Prediction Using Deep-Learning Models” International Conference on Computational Intelligence and Data Science (ICCIDS 2018)
12. Adil MOGHAR (2020), “Stock Market Prediction Using LSTM Recurrent Neural Network” International Workshop on Statistical Methods and Artificial Intelligence
13. Yunus YETIS (2019), “Stock Market Prediction by Using Artificial Neural Network” Department of Electrical and Computer Engineering University of Texas at San Antonio
14. Sidra Mehtab (2020), “Stock Price Prediction Using CNN and LSTM Based Deep Learning Models” 2020 International Conference on Decision Aid Sciences and Application (DASA)
15. Vijay Krishna Menon (2022), “STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL” Centre for Computational Engineering and Networking (CEN)
16. G. Peter Zhang (2003), “Time series forecasting using a hybrid ARIMA and neural network model” Department of Management, J. Mack Robinson College of Business, Georgia State University
17. U. Thissen (2003), “Using support vector machines for time series prediction” Chemometrics and Intelligent Laboratory Systems
18. Dev Shah (2018), “A Comparative Study of LSTM and DNN for Stock Market Forecasting” 2018 IEEE International Conference on Big Data
19. Ferdiansyah (2019), “A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market” International Conference on Electrical Engineering and Computer Science (ICECOS) 2019
20. Prapanna Mondal (2014), “STUDY OF EFFECTIVENESS OF TIME SERIES MODELING (ARIMA) IN FORECASTING STOCK PRICES” International Journal of Computer Science, Engineering and Applications April 2014
21. Yong’an Zhang (2020), “A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM” School of Economics and Management, Beijing University of Technology
22. David M. Q. Nelson (2017), “Stock Market’s Price Movement Prediction With LSTM Neural Networks” Departamento de Ciencia da Computac¸ ao (DCC)
23. Yujin Baek (2018), “ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module” a Department of Financial Engineering, Ajou University, Worldcupro
24. Sidra Mehtab (2019), “A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing” School of Computing and Analytics NSHM Knowledge Campus
25. Li-Chen Cheng (2018), “Applied attention-based LSTM neural networks in stock prediction” 2018 IEEE International Conference on Big Data (Big Data)
26. Paulo Rotela Junior (2014), “An Applied Time Series Forecasting Model for the Bovespa Stock Index” Institute of Production Engineering and Management, Federal University of Itajuba
27. Khaled A. Althelaya (2018), “Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction” 2018 9th International Conference on Information and Communication Systems (ICICS)
28. Jian Cao (2019), “Financial time series forecasting model based on CEEMDAN and LSTM” School of Electronic and Information Engineering, Sichuan University
29. Pushpendu Ghosh (2020), “Forecasting directional movements of stock prices for intraday trading using LSTM and random forests” a Department of Computer Science & Information Systems, BITS Pilani
30. Siyu Yao (2018), “High-Frequency Stock Trend Forecast Using LSTM Model” The 13th International Conference on Computer Science & Education (ICCSE 2018)
31. Yifeng Wang (2018), “LSTM model optimization on stock price forecasting” 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science
32. Ishita Parmar (2018), “Stock Market Prediction Using Machine Learning” 2018 First International Conference on Secure Cyber Computing and Communication(ICSCCC)
33. Anita Yadav (2019), “Optimizing LSTM for time series prediction in Indian stock market” International Conference on Computational Intelligence and Data Science (ICCIDS 2019)
34. Shotaro Minami (2018), “Predicting Equity Price with Corporate Action Events Using LSTM-RNN” Journal of Mathematical Finance, 2018
35. Zhigang Jin (2020), “Stock closing price prediction based on sentiment analysis and LSTM” Neural Computing and Applications (2020)

**APPENDIX B: RESEARCH PROPOSAL**

**Abstract**

The stock market is considered to be very dynamic and complex in nature. Predicting the future stock price can be a challenging task as there are many factors involved making the stock price labile. These factors can be physical or psychological. Throughout these years various approaches have been put forward to predict the stock price including regression models, time series forecasting, support vector machines and neural networks. Among various neural networks, long short-term memory (LSTM) models can dispense state of the art predictions with proper adjustments of various parameters. This research proposes a comparative study between a Vanilla LSTM model and a Bi-Directional LSTM model. The models are to be assessed upon freely accessible data from yahoo finance including the historical stock prices of 3 stocks of National Stock Exchange of India.

1. Background

Investing is an act of committing money or capital now in the expectation of gaining more money in the future. Investing can be a means to become financially independent or it can help in preserving the accumulated wealth. Investing in the stock market is probably the most common way to invest, because of its advantages and charm.

There are stock exchanges all around the world. Investors or traders connect with the stock exchanges via their brokers to place buys or sell shares. Stock price prediction is a step further in making a valuable investment. While there are people that think that it might not be possible to predict future stock prices, however in the past years, researchers have evidently proved that it is possible to forecast future stock prices with high accuracy (Mehtab, 2020). More and more papers are being published with various methods of prediction involving regression models, time series forecasting and deep learning models. Among these methods, the most standardly utilized method is Artificial Neural Network (ANN) (Maswood 2020). ANNs are composed of a collection of artificial neurons, which loosely replicate the neurons in a biological brain. Although ANNs have shown acceptable results as they are capable of learning any nonlinear function, they do suffer from an over-fitting issue. However, Recurrent Neural Network (RNN) and especially LSTM models have an advantage over ANNs as they are capable of grasping long term dependencies in data. LSTM models have dispensed state of the art predictions with proper hyper-parameter tuning. This study proposes a prediction framework that can forecast future stock prices utilizing two LSTM models; Vanilla LSTM and Bi-Directional LSTM.

2. Problem Statement

Financial time series data has always been more complex and prone to errors than any other statistical data. There are two traditional approaches to forecast future stock prices; Technical analysis and Fundamental analysis (Kim Soon, 2013). Fundamental analysis uses revenues, future deals, profit margins and other factors to forecast future stock prices, whereas, technical analysis can be defined as a time series analysis to forecast future stock prices using the historical stock price data. Through the years, combination machine learning models such as SVMs, ANNs and RNNs have revealed complex non-linear patterns that are impossible to detect with linear algorithms (Moghar, 2020). These models have proved to be more effective in predicting the closing stock price than the linear regression models.

A feedforward neural network is the most elementary model used for stock price prediction (Kim Soon, 2013). The feedforward NN was the most fundamental type of ANN devised. Herein connections between the nodes do not form a cycle. A feedforward NN was used by (Morris, 2007) to predict the stock market index trading signals.

RNNs are considered to be good at modelling and processing sequential data, which is quite suitable for stock price prediction. RNN networks are popularly used on financial time series data for making predictions. RNNs take information from prior inputs to influence the current input and output. A CRNN forecasting model was proposed by (Wang, 2018) to predict future stock price of 9 forex pairs.

A LSTM model is a special kind of RNN that overcomes the problem of exploding gradient which occurs in traditional RNNs (Ma, 2020). LSTMs are capable of learning long term dependencies in time series data. A basic LSTM architecture consists of:

• a cell

• an input gate

• an output gate

• a forget gate

A comparative study of LSTM and Bi-Directional LSTM model was published by (Maswood, 2020), proposing that Bi-Directional LSTM yields lower root mean squared error (RMSE) than a traditional LSTM.

3. Research Question

How precisely can we forecast the future stock prices using a Vanilla LSTM and a Bi-Directional LSTM with proper hyperparameter tuning?

4. Aim and Objectives

This research aims to propose a stock price prediction framework that can forecast future stock prices with proper accuracy. The goal of this study is to forecast the future stock prices using a Vanilla LSTM and a Bi-Directional LSTM.

The research objectives are as follows:

• To study state of the art prediction frameworks used for forecasting future stock prices.

• To compare the Vanilla LSTM and Bi-Directional LSTM models by varying the number of epochs, hidden layers, dense layers and other units.

• To evaluate the performance of the proposed models based on appropriate error metrics.

5. Significance of the Study

The motivation behind this research comes from my work at Bennett Coleman and Co. Ltd. (Times Group). From examining business performance to spearheading market share analysis, my work also includes forecasting monthly revenue projections across various business segments that helps the sales team to push the de-growing agents. This study can be used in future literature review involving Vanilla LSTM and Bi-Directional LSTM.

6. Scope of the Study

This study proposes a prediction framework based on the Vanilla LSTM and Bi-Directional LSTM models to forecast future stock prices of NSE of India. The models are to be assessed upon the historical data of 3 stocks having a time span of 6 months.

7. Research Methodology

Through the years, combination of statistical models and machine learning models such as ANNs, RNNs, and SVMs have revealed complex non-linear patterns of financial time series data that are difficult to detect with linear algorithms. These models have proved to be more effective in forecasting the closing stock price than the linear regression models.

RNN networks are popularly used on financial time series data for making predictions. The reason for their popularity in time series application is that RNNs are commonly utilized to perform tasks involving sequential input data such as financial time series. RNN is a type of neural network that keeps a memory of what it has already processed and thus can learn from previous iterations during its training. RNNs are composed of three elementary components: the input layer, the hidden layers, and the output layer. Each layer is composed of nodes. RNNs have proved to dispense state of the art predictions through the years. A CRNN forecasting model was proposed by (Wang, 2018) to predict future stock price of 9 forex pairs. (Song, 2019) proposed a multi-task RNN for stock price prediction.

Despite being quite popular among time series analysis, RNNs do suffer from an exploding gradient problem (Fayeem, 2022). LSTMs overcome the exploding gradient problem as they are capable of grasping long term dependencies in historical time series data. (Maswood, 2020) has proposed a comparative study between a LSTM model and a Bi-Directional LSTM model. This study proposes a comparative study of two LSTM models; Vanilla LSTM and Bi-Directional LSTM. The models are to be assessed upon historical stock price and to be evaluated with appropriate error metrics.

The historical time series data of various stocks can be acquired from yahoo finance. For this study I have extracted historical data of NSE stocks namely; Tata Motors, Reliance and HDFC bank for the period of 09/08/2021 to 08/02/2022.

Before the historical time series can be modelled, it must be pre-processed. The data is split into train set (80%) and test set (20%). The LSTM models generate a function that can map a series of previous observations as input to an output observation. We can divide the sequence into multiple input/output patterns called samples. Here each sample has a specified number of time steps as input and just one time step as output.

A Vanilla LSTM is a LSTM model that has a single hidden layer of LSTM units, and an output layer used to make a prediction. This variant of LSTM is less common in the literature. The illustration of Vanilla LSTM is as follows (Yuan, 2018).

Vanilla LSTM

In short, LSTM has the ability to add or delete information to the cell state. This can be regulated by the various gates present in LSTM namely; input, output and forget gate.

In a Bi-Directional LSTM inputs flow in two directions; forward (past to future) and backwards (future to past), making a Bi-Directional LSTM different from a common LSTM. Bi-Directional LSTMs effectively increase the amount of information available to the network. With this form of network, the output layer can get information from past and future states simultaneously. The illustration is as follows:

Bi-Directional LSTM

The models are to be implemented using some common python libraries such as Numpy, Pandas, Tensorflow, Keras and Matplotlib via Jupyter Notebook or Google Collab. TensorFlow is a Python library for fast numerical computing whereas keras provides an interface for ANNs.

The models are then to be tested and evaluated by tuning the number of epochs, hidden layers, dense layers and other units. The models can then be compared based on the predictions with suitable error metrics such RMSE or MAPE and on the time taken for the predictions.

8. Resources Required

The software requirements are as follows:

• Jupyter Notebook or Google Collab

• Python libraries such as Numpy, Pandas, Tensorflow, Keras, Matplotlib

The hardware requirements are as follows:

• 1GB RAM + 1GB of disk + .5 CPU core.

9. Research Plan

The research plan is illustrated as a Gantt chart below:

The possible risks encountered can be

• The python ide (Jupyter Notebook) can crash if the models are too complex to execute. The code can then be executed on platforms such as Google Collab and Tensorflow.

References

1. Sidra Mehtab (2020), “Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models” 2020 International Conference on Decision Aid Sciences and Application (DASA)

2. Mirza Mohd Shahriar Maswood (2020), “Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model” 2nd Novel Intelligent and Leading Emerging Sciences Conference

3. Gan Kim Soon (2013), “A Review of Stock Market Prediction with Artificial Neural Network (ANN)” 2013 IEEE International Conference on Control System, Computing and Engineering

4. Adil Moghar (2020), “Stock Market Prediction Using LSTM Recurrent Neural Network” International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI 2020)

5. S. A. Morris (2007), “Predicting Stock Market Index Trading Signals Using Neural Networks” Centre for Informatics and Applied Optimization School of Information Technology and Mathematical Sciences University of Ballarat

6. Xiao Wang (2018), “Forecasting of Forex Time Series Data Based on Deep Learning” 2018 International Conference on Identification, Information and Knowledge in the Internet of Things

7. Qihang Ma (2020), “Comparison of ARIMA, ANN and LSTM for Stock Price Prediction” E3S Web of Conferences 218, 01026 (2020)

8. Dongjin Song (2019), “Multi-task Recurrent Neural Networks and Higher-order Markov Random Fields for Stock Price Movement Prediction” KDD ’19, August 4–8, 2019, Anchorage, AK, USA

9. Abdul Fayeem (2022), “Stock Price Prediction : Recurrent Neural Network in Financial Market” International Journal for Modern Trends in Science and Technology · January 2022