Stock Price Prediction using Bi-Directional LSTM based Sequence to Sequence Modeling and Multitask Learning

Siddartha Mootha
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
siddartha.mootha20@gmail.com

Sashank Sridhar
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
sashank.ssridhar@gmail.com

Rahul Seetharaman
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
rahulseetharaman@gmail.com

Dr. S. Chitrakala
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
chitrakala.au@gmail.com

***Abstract*— The stock market is a dynamic and volatile platform which provides an environment for traders to invest and trade in shares. The price of a stock is dependent on numerous static and dynamic features. Predicting the future price of a particular company's stock can be extremely beneficial for traders. Seq2Seq modelling helps map an input sequence to an output sequence. In this paper, we propose a system to predict the future Open, High, Close, Low (OHCL) value of a stock using a Bi-Directional LSTM based Sequence to Sequence Modelling. Each OHCL price is an independent sequence and multitask learning helps map the interrelations between them. A multitask system is also proposed which uses sub tasks and shared tasks to model the prices. Stock prices of Tata Consumer Products Limited from the National Stock Exchange (NSE) of India is used. To evaluate the efficiency of the proposed systems, they are compared against various machine learning algorithms. The proposed Seq2Seq and multitask systems comfortably outperform the existing algorithms with RMSE values of 3.98 and 7.87 respectively.**

Keywords—Stock Forecasting, Sequence to Sequence Modelling, Multitask Learning, Bi-directional LSTM, Prediction System

I. Introduction

The stock market is an extremely potent environment capable of handling millions of transactions in a short span of time [1]. Every transaction is routed through a central exchange [2]. The stock market involves three primary parties i.e. a buyer, seller and an exchange [3]. Generally a buyer purchases a stock via a third party agent such as a broker [4]. A stock is a measure of ownership of a company [5].

There are different types of companies [6] such as statutory companies, Royal Chartered Companies, Private Limited Companies, Public Limited Companies etc. When a company decides to offer shares of stock to the general public, it must launch an Initial Public Offering (IPO) [7]. This offering states the number of shares available and the price at which it is available. Although the price of a stock is extremely volatile, it can change drastically from time to time.

The stock market is not static, it is an extremely dynamic environment [8]. The market is non-linear and can be highly unpredictable [9]. The financial market provides an environment for traders to invest and trade in shares, stocks, securities and equities. The stock market provides an environment where a small investment could lead to a large return, or even a large loss. Predicting the future price [10] of a particular company's stock could be extremely beneficial for investors, nonetheless it is extremely challenging. The price of a stock varies on numerable aspects [11] including but not limited to weather forecasts, speech by a politician, social media, news articles, global economy etc. As can be seen, most of the above mentioned attributes are not controllable by an average human. Nonetheless, with the advancement in artificial intelligence and machine learning, and the availability of high performance super computers, predicting the future price of a stock is possible [12].

A general stock has 4 primary price features [13], i.e. Open Price, High Price, Low Price, Close Price (OHLC). Based on these 4 primary price attributes, this paper proposes various systems to predict the future price of a stock.

Sequence to Sequence (Seq2Seq) [14] modelling helps map an input sequence of data to an output sequence of data. The Seq2Seq models use an encoder-decoder framework [15]. The encoder helps map the structure of the input sequence and the decoder generates the corresponding output sequence. The encoder and decoder make use of Recurrent Neural Networks (RNN) [16] for analysing the sequence of the data. Bi-directional LSTMs (Bi-LSTM) [17] are a form of RNNs that map the sequences in both forward and reverse directions. This helps the network deal with increased sequence length and long term dependencies.

A Bi-directional LSTM (Bi-LSTM) model which maps the input sequence of OHLC prices for a particular day is proposed in this paper. The proposed Seq2Seq model aims to generate an output sequence of OHLC data for the day under study. In stock price prediction, the future price is as important as the past price. By creating a window based model [18], the past and future prices are modelled using Bi-LSTMs.

978-1-7281-9656-5/20/$31.00 ©2020 IEEE

Multitask learning [19] is a machine learning paradigm that works effectively if the parameter space consists of parameters which are interrelated. It is similar to how humans learn something. For example, when humans learn to ride a car, they understand how to use gears, clutches, accelerators, etc. This knowledge can help them to learn to ride a motorbike as well. It can improve performance by learning to solve overlapping subproblems and using the knowledge obtained in solving one sub-problem to effectively use it for another sub-problem, in a joint learning process [20].

In this paper, the multitask model is established to determine the variation of OHLC prices. By determining the relative changes in OHLC prices, the prediction error can be reduced. Multitask learning helps identify the interrelation between the OHLC prices by predicting the OHLC prices simultaneously. As the prediction of each of the OHLC prices is viewed as an independent task and as the OHLC prices are dependent on one another, this forms a multitask learning problem.

To test the performance of the system, the two models are compared against various algorithms such as k-NN, Bayesian Regression, Ridge Regression, Linear Regression, Huber Regression, TheilSen Regression, Lasso Regression, Least Angle Regression and SVM. The results of the comparison show how well the proposed models perform by outperforming the basic Machine Learning models.

The rest of the paper is organized as follows: In Section II, we will discuss the related works carried out in stock price prediction. Section III explains about the design of the system in detail. Section IV talks about the implementation details. Section V analyses the results. In Section VI, the conclusion and future works are discussed.

II. Related Works

Stock price prediction can be done using Artificial Neural Networks (ANN). ANNs use adaptive weights to forecast stock prices. Y. Bing et al. [21] proposed an ANN to predict the index of the Shanghai Stock Exchange. The authors studied the market between March 17, 2010 to April 28, 2010. They considered 5 features of the market, open, high, close, low and volume. The neural network constructed was successful in predicting the daily lowest, highest, and closing value of the Shanghai Stock Exchange. The gradient search technique and learning algorithm were constructed in the deep learning model. The results showed that the model was successful in predicting the daily low, high, and closing value of the Shanghai Stock Exchange for a short time period, but inefficient in predicting the return rate of the Shanghai Stock Exchange.

RNNs help identify the relationship present in sequential data. Bi-LSTMs are a subset of RNNs which model the data in forward and reverse directions. This takes into account what the past stock prices were and what the future stock prices might be. M. Jia et al. [22] proposed a framework which made use of the bidirectional long-short term memory (BLSTM) neural network for predicting the future price of a stock. The authors used the historical data of the GREE stock. They collected data for 568 days from January 1, 2017 to May 14, 2019. The data consisted of 14 features such as open, high, close, volume etc. The data was normalized and pre-processed. The close value was used as the benchmark for the prediction. The pre-processed data was put into a one way and two way LSTM. Dropout is an optimization technique to prevent overfitting. Dropout technique was applied on the neural network. The proposed BLSTM was compared against the ARIMA model as well as a traditional Long Short Term Memory (LSTM). To measure the performance of the proposed system, RMSE, MAE and deviation accuracy was calculated. It could be seen from the results that the proposed system outperformed the ARIMA model as well as the LSTM model. RMSE and MAE were reduced by 24.2% and 19.4% respectively.

K. A. Althelaya et al. [17] proposed a Bidirectional LSTM for Short- and Long-Term Stock Market Prediction. The authors had made use of the Standard and Poor 500 Index (S&P500) historical data for their proposed work. The S&P500 is a leading indicator for various top traded companies. The dataset was normalised and pre-processed. At the end of every trading day, there is a closing price. The closing price was used for the sake of evaluating the performance of the system. The authors proposed two systems, one for short term prediction and one for long term prediction. To evaluate the performance of both the models, it was evaluated and compared against 3 different models, i.e. MLP-ANN, LSTM and SLSTM. The BLSTM achieved a lower RMSE and MAE score in comparison to the other models. The BLSTM outperformed the existing models and achieved better convergence for short term and long term predictions.

Sequence to sequence modelling helps the network map a sequential input to a corresponding sequential output. C.-H. Cho et al. [23] modelled the Taiwan stock price information of 932 companies in the period between 2007 and 2019. They trained three different models : LSTM, Seq2Seq and WaveNet using daily trading data as well as technical indicators. They found the correlation between the predicted price and actual price and used the correlation as a performance metric. The WaveNet model proposed by them outperformed the other two models.

Every stock exchange has an index. The market index is an extremely critical factor for investors. J. Eapen et al. [24] proposed a deep learning model with Convolutional Neural Networks (CNN) and Bi-Directional LSTM (Bi-LSTM) to predict the future index of the market. The proposed model was implemented on the S&P 500 grand challenge dataset. Multiple pipelines of CNN and BLSTM units were combined. The authors presented various combinations of single and multiple pipeline deep learning models applied on various kernel sizes for CNN and for varying numbers of Bi-LSTM units. It was observed that the deep network created due to CNNs, when combined with Bi-LSTM created a more efficient system than previously existing ones.

As shown by J. Eapen et al. [24], Bi-LSTMs increase the accuracy of prediction by mapping the sequential data in both directions. In our first model, we have set up a Seq2Seq model which uses Bi-LSTMs to map the input sequential data.

C. Li et al. [25] built a multi-task RNN to extract features from OHLC, volume and amount data of a particular stock. Features extracted from each individual task were then concatenated and mapped to predict future prices. The multi-task model was a “MultiTask Market Price Learner” and

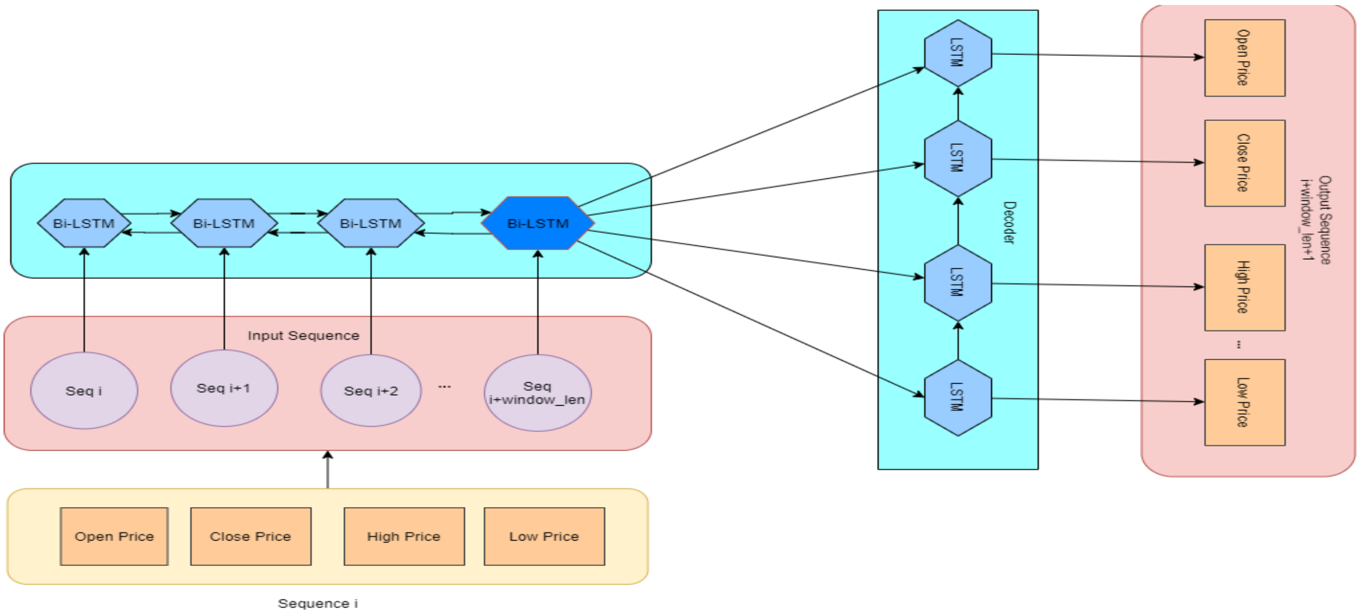

Fig. 1. Bi-Directional LSTM based Encoder-Decoder

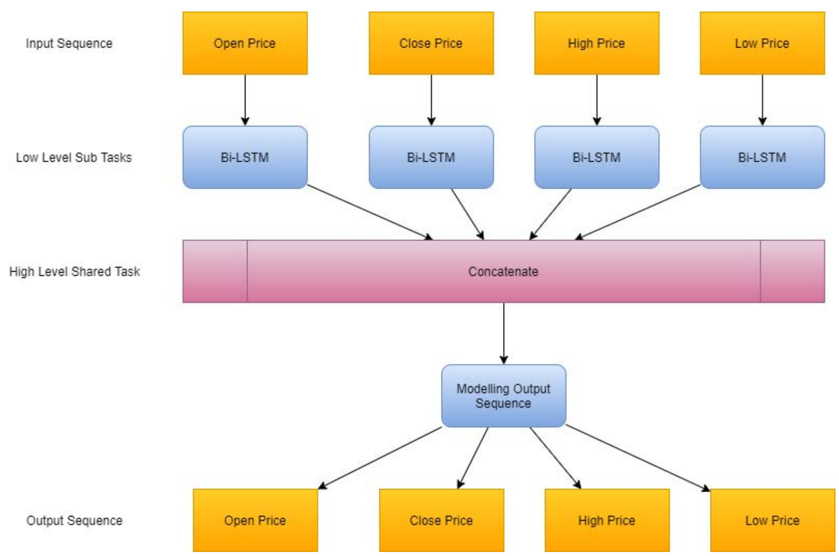

Fig. 2. Multitask Learning Model

consisted of three dual stage attention based recurrent neural networks (DARNN). DARNNs help extract temporal features and were shown to perform better than LSTMs and attention based LSTMs. Two types of tasks were handled by the model proposed by the authors. The low-level tasks were used to predict volatility and stock prices in the future. The higher level task made use of the low-level tasks to predict the movement of stock prices. The model was trained on data from the CSI 200, 300 and 500 indices of the Chinese stock market. The Technical Analysis Library in python was used to extract 8 technical indicators for each stock. These features along with the market features i.e. OHLC, Volume and Amount, were trained using the multi-task learner. The multi-task learning model achieved an accuracy of 63.67% for trends and 62.50% for volatility.

It can be seen from C. Li et al. [25] that multitask learning helps model individual tasks and the overall interrelation between the tasks. In our second model, we use multitask learning to model the OHLC data as individual tasks and to predict the future OHLC values. As we use multitask learning the variations in OHLC prices are interdependent on each other.

The price of a stock can vary based on factors such as blogs, news, articles, rumours etc. The numerous external factors make it extremely difficult to predict the price of a stock. W. Khan et al. [26] proposed a system to predict the market index as well as the price of a stock based on social media news. Impact of social media and financial news on a stock is studied. Deep learning and an ensemble of classifiers is used for prediction. The results show that social media influences the New York and IBM stocks the most, whereas financial news influences the London and Microsoft stocks the most.

It is observed from W. Khan et al. [26] that stock prices are affected by multiple factors. By using multitask learning, we try to model the impact of each of the OHLC prices on the overall stock price, individually. The lower level tasks in our model maps the individual OHLC prices and the higher level task forecasts the OHLC prices.

III. Design Methodology

A. Dataset Description

Stock prices for every company listed on a stock exchange is readily available. This paper makes use of the stock prices of Tata Consumer Products Limited from the National Stock Exchange (NSE) of India [27]. The stock prices were extracted in the period between 2013 and 2018. The dataset contains four main features - Open, High, Low and Close Price (OHLC) as seen in Table I.

TABLE I. DATA ATTRIBUTES

S. No.	Feature	Description
1	Open Price	Opening price of the stock on a particular day
2	Close Price	The closing price of the stock at the end of a trading day
3	High Price	The peak price on a particular day
4	Low Price	The lowest price of the stock on a particular day

B. Sequence to Sequence Modelling

Sequence to sequence modelling tries to forecast the consecutive output of the given input sequence [28]. Seq2Seq modelling tries to predict the output sequence using a given input sequence [29]. The problem solved by Seq2Seq mapping is that sequences of variable length can be mapped to the output sequence. Seq2Seq models can be used in a variety of tasks, mainly those that require generation of text, sequences, etc [30].

C. Bi-Directional LSTM based Encoder-Decoder

Encoder-Decoder [31] is a method to solve Seq2Seq predictions. The encoder converts a variable length input sequence to a fixed-length vector which is decoded by the decoder. The decoder tries to predict the output sequence by using the encoded input. This method helps map variable length sequences.

Figure 1 shows the overall architecture diagram of the implemented Seq2Seq model. OHLC prices of stocks for each day form a sequence. An input sequence is taken as a sequence of OHLC prices for a particular window length of days. The output sequence is the sequence of OHLC prices for the next day after the input window.

The encoder consists of Bi-LSTM units, each of which maps the Open, Close, High and Low price. The Bi-LSTM units try to model the relation between the past sequences and the future sequences. At a particular instant, a sequence of window length units are given to the encoder. The window length is chosen such that the output prediction error is minimal. The encoder converts the input sequence to a fixed length vector. This vector is decoded by the LSTM units of the decoder. The decoder tries to predict a single sequence of OHLC prices.

D. Multitask Learning

Multitask learning involves learning sub tasks and training a common shared task using the information learnt [32]. In the implemented system as seen in Figure 2, each of the OHLC prices is taken as a subtask. Input sequence is prepared using a sliding window of a particular window length. The window length is chosen such that the loss is minimal. Sequences of length n for each price are trained with the output being the $n + 1^{th}$ price in the sequence. Each sub task is modelled using a Bi-LSTM model. The outputs of the individual sub tasks are concatenated and given to a shared Bi-LSTM model which predicts the output sequence for each of the OHLC prices. In this way the features learnt from each sub task is propagated to a common shared task which is responsible for determining the output.

IV. Implementation

Implementation of the proposed models was done on a Windows 10 machine with Intel I7 8th Gen Processor, 16 GB RAM and NVidia GEForce GTX 1060 graphics processor.

A. Preprocessing the Dataset

The dataset collected has OHLC prices for a total of 1150 samples. The dataset was split into a training and testing set with a ratio of 75:25. The dataset is cleaned to remove any null values. The data is then normalized using the MinMaxScaler of sklearn [33]. This scaling subtracts the minimum value in the dataset from every value in the dataset, and divides every value in the dataset by the maximum value in the dataset. This procedure was repeated across multiple time series, each of which denoted Opening Price, Closing Price, High Price, and Low Price.

B. Determining the Sliding Window

A sequence of OHLC prices was created by using a sliding window. For all the values at every time step, a window size of i was chosen as the input and the next value after it, $i + 1^{th}$, was chosen as the output. This procedure was repeated as part of a sliding window procedure and the input and output was generated, which was in turn fed to both the constructed models.

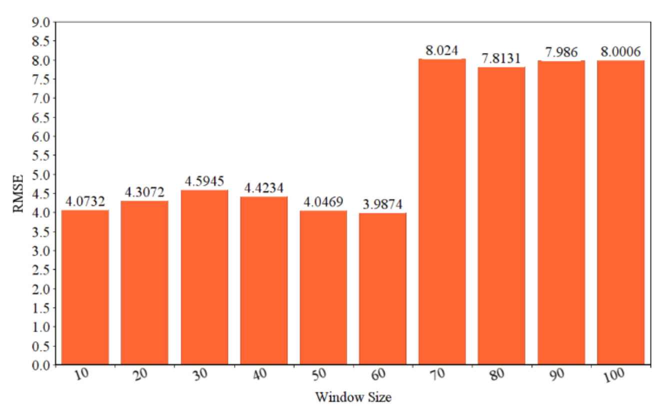

Fig. 3. Variation of RMSE with Window Size in Bi-LSTM based Seq2Seq Model

From Figure 3, it can be seen that the RMSE is lowest at a window size of 60 for the Bi-LSTM based Sequence to Sequence model.

Algorithm 1 shows how the dataset is created for the Seq2Seq model. The training set is taken as a sequence of determined window size. The training outputs have all 4 of the OHLC prices together.

Algorithm 1 : Dataset for Seq2Seq model

```
Function Seq2Seq():
    data ← MinMaxScaler(data)
    for i in range(window_length, len(data)) do
        train_x.append(data[i-window_length:i,:]
        train_y.append(data[i-1:i,:]
    end
end
```

From Figure 4, it can be seen that the RMSE is lowest at a window size of 60 for the Bi-LSTM based Multitask model.

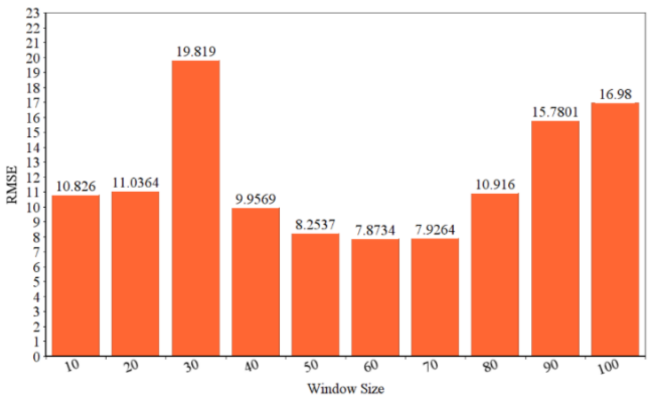

Fig. 4. Variation of RMSE with Window Size in Bi-LSTM based Multitask Model

Algorithm 2 : Dataset for Multitask model

```
Function Multitask():
    data ← MinMaxScaler(data)
    for i in range(window_length, len(data)) do
        train_x.append(data[i-window_length:i,:]
        train_y_open.append(data[i,0])
        train_y_close.append(data[i,1])
        train_y_high.append(data[i,2])
        train_y_low.append(data[i,3])
    end
end
```

Algorithm 2 shows how the dataset is created for the Multitask model. The training set is taken as a sequence of determined window size. The training outputs are segregated separately for OHLC prices. This is because prediction of each price forms a sub task in the multitask model.

C. *Bi-LSTM based Seq2Seq Model*

Figure 5 shows the implementation of the Seq2Seq model. The Bi-LSTM based Seq2Seq model makes use of an encoder-decoder architecture. The encoder is Bi-directional LSTM layer with 256 nodes. Figure 6 shows the variation of RMSE with the number of neurons in the encoder. It can be seen that the RMSE is lowest when the number of neurons in the encoder is 256.

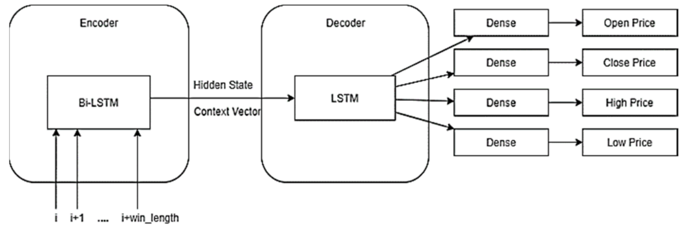

Fig. 5. Implementation of Seq2Seq model

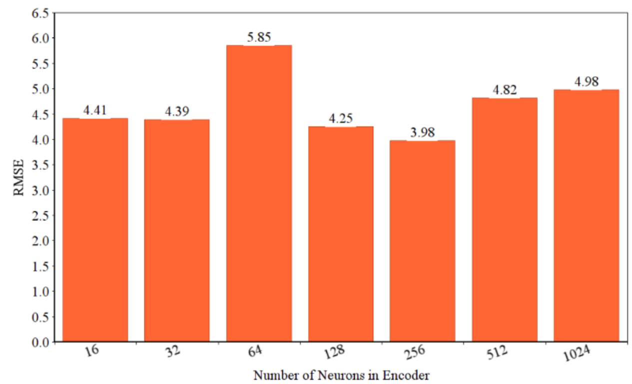

Fig. 6. Variation of RMSE with Number of Neurons in the Encoder

The encoder returns two vectors [34] of fixed size. One is a hidden state vector and another is a context vector. The decoder is a LSTM layer with 512 nodes. The decoder takes the output states of the encoder and returns a sequence of output values. A dense layer is used to map the decoder outputs to the OHLC values of the output sequence. The Seq2Seq model is trained for 10 epochs.

D. *Bi-LSTM based Multitask Model*

Figure 7 shows the overall implementation of the multitask model. In the multitask model, each sub task is a Bi-LSTM model that tries to learn each of the OHLC prices. Each sub task has 2 Bi-LSTM layers with 128 and 64 nodes respectively. From Figure 8 it can be seen that the RMSE is lowest at 2 hidden layers. Each Bi-LSTM layer is followed by a dropout of 0.2.

The final output from the 4 sub tasks is a 4 * 1 dimensional vector that predicts open, close, high and low prices. This output vector is fed to a shared task that tries to reduce the loss of predictions further. This shared task has 2 Dense layers with 100 and 50 nodes each. It can be seen from Figure 9 that the RMSE is lowest when the shared task has 2 hidden layers.The shared task is mapped to the OHLC prices using 4 output Dense layers as seen in Figure 7.The multitask model is trained for 5 epochs.

E. *Parameters for Training*

Table II shows the parameters used for training the models. The optimizer used is RMSProp [35]. RMSProp maintains a moving average of the square of gradient values and divides the gradient by the root of the moving average. A learning rate of 0.001 was used for training the model over the generated data. The loss function used was the mean squared error loss function

[36], which computes the mean of the squared difference between the label and predictions. The squared difference is done to avoid positive and negative loss differences being nullified across the dataset.

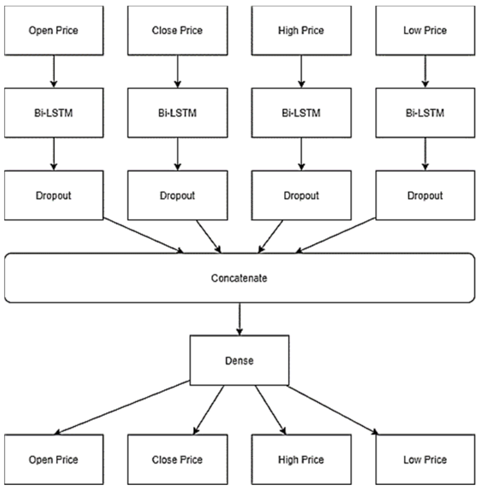

Fig. 7. Implementation of Multitask model

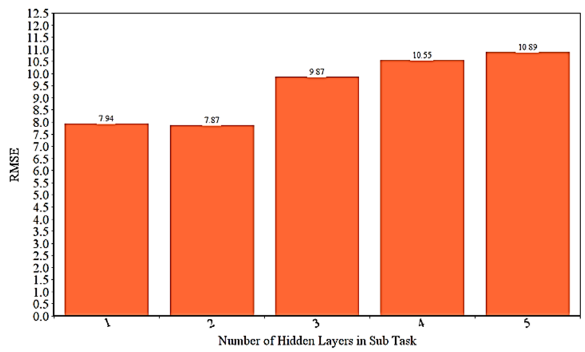

Fig. 8. Variation of RMSE with Number of Hidden Layers for Sub Tasks

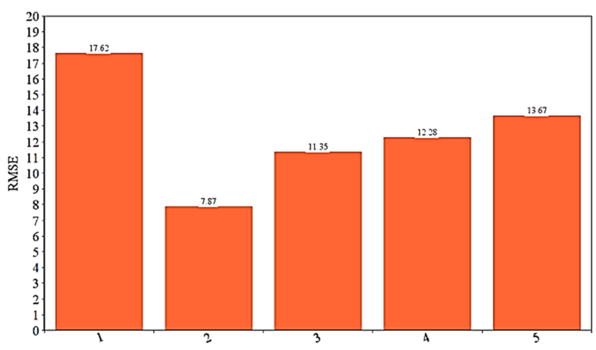

Fig. 9. Variation of RMSE with Number of Hidden Layers in Shared Task

TABLE II. PARAMETERS FOR TRAINING

S. No.	Parameter	Value
1	Optimizer	RMSProp
2	Loss Function	Mean Squared Error
3	Learning Rate	0.001
4	Batch Size	32

V. RESULTS AND ANALYSIS

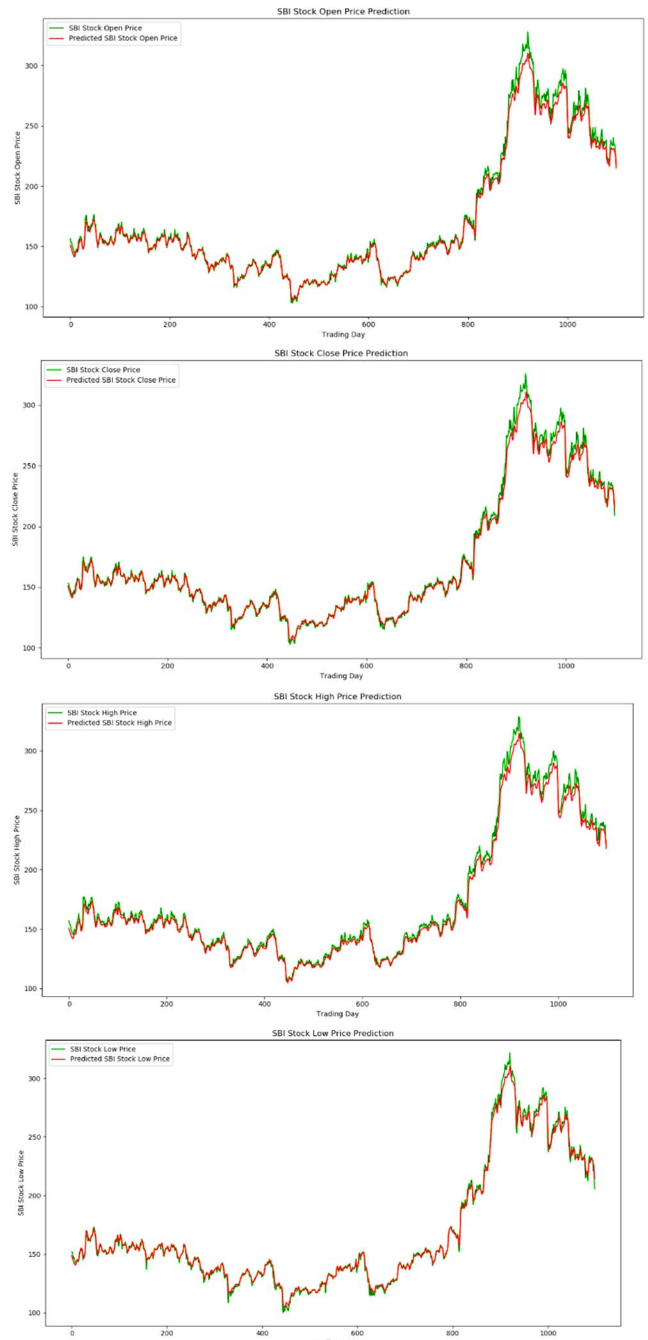

Fig. 10. Prediction curves for a) Open b) Close c) High and d) Low prices by Bi-LSTM based Seq2Seq model

Figures 10 and 11 show the prediction curves of OHLC data for both the models. The green line shows the actual data and the orange line shows the predicted data. The closeness of the two lines determines how well the models predict the outputs. For any machine learning problem, it is rudimentary and critical to evaluate the performance of the proposed model. The proposed system is evaluated using the following metrics where 'n' represents the number of observations, y_i represents the predicted price of the stock, and x_i represents the actual value of the stock.

Fig. 11. Prediction curves for a) Open b) Close c) High and d) Low prices by Bi-LSTM based Multitask model

$$Mean\ Absolute\ Error\ (MAE) = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (1)$$

$$Mean\ Squared\ Error\ (MSE) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \quad (2)$$

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \quad (3)$$

The proposed seq2seq and Bi-LSTM multitask model is compared against existing regression models. In Machine Learning, regression [38] is a paradigm where future values are predicted by studying and learning the trends of the existing values. The proposed system is compared against regression models such as k-NN Regressor, Bayesian Ridge Regressor, Ridge Regression, Linear Regression, Huber Regression, TheilSen Regressor, Random Sample Consensus, Orthogonal Matching Pursuit, Lasso Regression, SVM Regression, Passive Aggressive Regressor, Elastic Net, Lasso Least Angle Regressor and Least Angle Regression. The above-mentioned models are implemented using the ‘pycaret’ [37] library available in python. Ten-fold cross validation is performed to ensure that the models do not overfit.

k-NN [39] is a powerful algorithm which can be used for classification and regression cases. In regression, the algorithm makes use of feature similarity to determine the nearest neighbours. The nearest neighbours can be found using a distance formula such as Euclidean or Manhattan distance. k represents the number of neighbours chosen. The model was tested for varying K values, the system achieved its lowest RMSE of 12.17 at 5 neighbours.

The Bayesian Ridge [40] is a probabilistic regression algorithm. The algorithm makes use of probabilistic distributors over classical point estimators. The algorithm is able to tackle datasets that are poorly distributed or have insufficient data. The algorithm achieved an RMSE of 13.13. The loss function for the ridge regression algorithm is the linear least square function, using this function, the Ridge Regression achieved an RMSE of 13.14.

Linear regression [41] is a statistical model that creates a linear relationship between the features of the dataset and the class label. The input features are fed as a linear combination, along with the class label. The model achieved an RMSE of 13.19.

The Huber regression [42] makes use of the Huber loss function, which ensures that the model is not heavily affected by outliers in the dataset, the Huber regression algorithm achieved an RMSE of 13.4.

TheilSen regression [43] is another statistical algorithm that computes the best fit line by calculating the median of the slopes. The algorithm is efficient in tackling random outliers in the dataset. The algorithm achieved an RMSE of 13.57.

Lasso [44] is an acronym that means Least Absolute Shrinkage and Selection Operator. As the name suggests, the lasso regression algorithm makes use of shrinkage. All the data points are shrunk to a value, which can be the mean or the central value. Using this, the future values are predicted. The algorithm achieved an RMSE of 14.58.

SVM [45] is a supervised algorithm which can be used for classification and regression. The algorithm constructs a hyperplane and a decision boundary. Any value that lies outside the decision boundary is an outlier, the hyperplane acts as the best fit line for the model. The linear kernel is used. The algorithm achieved an RMSE of 14.90.

If the linear regression model needs to be developed to work on a higher dimension dataset, the least angle algorithm [46] can be made use. The algorithm achieved an RMSE of 227.5, since the dataset is not of a higher dimension, the RMSE value is significantly high. By applying the LASSO algorithm on the least angle algorithm, shrinkage is possible. The lasso least angle regression achieved an RMSE of 18.6.

TABLE III. COMPARISON OF PROPOSED MODELS AGAINST MACHINE LEARNING MODELS

Model	MAE	MSE	RMSE
Bi-LSTM Seq2Seq Model	**2.6845**	**15.899**	**3.9874**
Bi-LSTM Multitask Learning	**5.4586**	**61.9907**	**7.8734**
k-Neighbours Regressor	9.2576	149.553	12.1733
Bayesian Ridge	9.7404	174.392	13.1391
Ridge Regression	9.7545	174.625	13.1467
Linear Regression	9.8032	175.807	13.1914
Huber Regressor	9.4407	181.94	13.4108
TheilSen Regressor	10.1028	186.304	13.5736
Random Sample Consensus	9.4097	188.366	13.6552
Orthogonal Matching Pursuit	10.2667	194.284	13.8447
Lasso Regression	11.1123	214.692	14.5814
Support Vector Machine	10.5572	224.368	14.9036
Passive Aggressive Regressor	10.8883	226.655	14.9515
Elastic Net	12.3847	276.343	16.5724
Lasso Least Angle Regressor	13.8716	348.526	18.6167
Least Angle Regression	151.758	159.069	227.599

As can be seem from Table III, the proposed Seq2Seq model and Bi-LSTM Multitask model comfortably outperform the existing regression models. A low RMSE indicates that the model is predicting the future value with accuracy. As can be seen from Table III, the two proposed models have a significantly low RMSE value in comparison to existing regression algorithms. This demonstrates that the proposed models are able to predict the upward and downward trend of the price of a stock.

VI. CONCLUSION AND FUTURE WORKS

In this paper, a system to predict the future Open, High, Close, Low (OHCL) price of a stock using a Bi-Directional LSTM based Sequence to Sequence Modelling has been proposed. The system is tested on the stock price of Tata Consumer Products Limited from the National Stock Exchange (NSE) of India. The Seq2Seq model and Multitask model achieved an RMSE value of 3.98 and 7.87 respectively. The proposed models were compared to various existing regression models, and outperformed them.

In the future, the system can be tested for a combination of static and dynamic features. Further, sentiment analysis can be performed on news articles, blogs, social media, etc, to determine the upward or downward trend of a stock.

REFERENCES

[1] B. Chhimwal and V. Bapat, “Impact of foreign and domestic investment in stock market volatility: Empirical evidence from India,” *Cogent Economics & Finance*, vol. 8, no. 1, Apr. 2020.

[2] S. Sridhar, S. Mootha, and S. Subramanian, “Decentralized Stock Exchange Implementation using Ethereum,” in *2020 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 234-241, 2020.

[3] C. Pop *et al.*, “Decentralizing the Stock Exchange using Blockchain An Ethereum-based implementation of the Bucharest Stock Exchange,” in *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 459-466, 2018.

[4] N. Sakthivel and A. Saravanakumar, “Investors’ Satisfaction on Online Share Trading and Technical Problems Faced by the Investors: A Study in Coimbatore District of Tamilnadu,” *International Journal of Management Studies*, vol. V, no. 3(9), p. 71, Jul. 2018.

[5] D. Shah, H. Isah, and F. Zulkernine, “Stock Market Analysis: A Review and Taxonomy of Prediction Techniques,” *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, May 2019.

[6] U. Hathi, “Indian Companies Act 2013 Highlights and Review,” *SSRN Electronic Journal*, 2014.

[7] S. V. Shenoy and K. Srinivasan, “Relationship of IPO Issue Price and Listing Day Returns with IPO Pricing Parameters,” *International Journal of Management Studies*, vol. V, no. 4(1), p. 11, Oct. 2018.

[8] G. Tanty and P. K. Patjoshi, “A Study on Stock Market Volatility Pattern of BSE and NSE in India,” *Asian Journal of Management*, vol. 7, no. 3, p. 193, 2016.

[9] S. Sridhar, S. Mootha, and S. Subramanian, “Detection of Market Manipulation using Ensemble Neural Networks,” in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1–8, 2020

[10] S. Zavadzki, M. Kleina, F. Drozda, and M. Marques, “Computational Intelligence Techniques Used for Stock Market Prediction: A Systematic Review,” *IEEE Latin America Transactions*, vol. 18, no. 04, pp. 744–755, Apr. 2020.

[11] M. C. Joshi, “Factors Affecting Indian Stock Market,” *SSRN Electronic Journal*, 2013.

[12] S. Alhazbi, A. B. Said and A. Al-Maadid, "Using Deep Learning to Predict Stock Movements Direction in Emerging Markets: The Case of Qatar Stock Exchange," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, pp. 440-444, 2020.

[13] D. Wei, “Prediction of Stock Price Based on LSTM Neural Network,” in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pp. 544-547, 2019.

[14] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, “Deep Reinforcement Learning for Sequence-to-Sequence Models,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2469–2489, 2019.

[15] K. Palasundram, N. Mohd Sharef, N. A. Nasharuddin, K. A. Kasmiran, and A. Azman, “Sequence to Sequence Model Performance for Education Chatbot,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 24, p. 56, Dec. 2019.

[16] H. Jain and G. Harit, "An Unsupervised Sequence-to-Sequence Autoencoder Based Human Action Scoring Model," *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Ottawa, ON, Canada, pp. 1-5, 2019.

[17] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, “Evaluation of bidirectional LSTM for short-and long-term stock market prediction,” in *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 151-156, 2018.

[18] J. Chou and T. Nguyen, "Forward Forecast of Stock Price Using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3132-3142, July 2018.

[19] Y. Zhang and Q. Yang, “An overview of multi-task learning,” *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.

[20] B. Ko and H. Choi, "Paraphrase Bidirectional Transformer with Multi-task Learning," *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea (South), pp. 217-220, 2020.

[21] Y. Bing, J. K. Hao, and S. C. Zhang, “Stock Market Prediction Using Artificial Neural Networks,” *Advanced Engineering Forum*, vol. 6–7, pp. 1055–1060, Sep. 2012.

[22] M. Jia, J. Huang, L. Pang, and Q. Zhao, “Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network,” in *Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*, pp.467-473, 2019.

[23] C.-H. Cho, G.-Y. Lee, Y.-L. Tsai, and K.-C. Lan, “Toward Stock Price Prediction using Deep Learning,” in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion - UCC ’19 Companion*, pp. 133–135, 2019.

[24] J. Eapen, D. Bein, and A. Verma, “Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0264-0270, 2019.

[25] C. Li, D. Song, and D. Tao, “Multi-task Recurrent Neural Networks and Higher-order Markov Random Fields for Stock Price Movement Prediction,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1141–1151, 2019.

[26] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, “Stock market prediction using machine learning classifiers and social media, news,” *Journal of Ambient Intelligence and Humanized Computing*, Mar. 2020.

[27] “TATA CONSUMER PRODUCTS LIMITED Share Price Today, Live NSE Stock Price, News – NSE India,” NSE India, 2020. [Online]. Available: https://www.nseindia.com/get-quotes/equity?symbol=TATACONSUM. [Accessed: 17-Sep-2020].

[28] S. Hwang, G. Jeon, J. Jeong, and J. Lee, “A Novel Time Series based Seq2Seq Model for Temperature Prediction in Firing Furnace Process,” *Procedia Computer Science*, vol. 155, pp. 19–26, 2019.

[29] A. Joshi, K. Mehta, N. Gupta and V. K. Valloli, "Data Generation Using Sequence-to-Sequence," *2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Thiruvananthapuram, India, pp. 108-112, 2018.

[30] G. Aalipour, P. Kumar, S. Aditham, T. Nguyen and A. Sood, "Applications of Sequence to Sequence Models for Technical Support Automation," *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, pp. 4861-4869, 2018.

[31] Y. Chen, W. Lin and J. Z. Wang, "A Dual-Attention-Based Stock Price Trend Prediction Model With Dual Features," in *IEEE Access*, vol. 7, pp. 148047-148058, 2019.

[32] G. Balikas, S. Moura, and M.-R. Amini, “Multitask Learning for Fine-Grained Twitter Sentiment Analysis,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1005–1008, 2017.

[33] “sklearn.preprocessing.MinMaxScaler — scikit-learn 0.22.1 documentation,” Scikit-learn.org, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[34] L. Lu, X. Zhang, K. Cho, and S. Renals, “A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition,” in *Proceedings of the Annual Conference of the International Speech Communication Association*, INTERSPEECH, pp. 3249–3253, 2015.

[35] S. V. G. Reddy, K. T. Reddy, and V. ValliKumari, “Optimization of Deep Learning using various Optimizers, Loss functions and Dropout,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 4S2, pp. 448–455, 2018.

[36] A. Botchkarev, “A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms,” *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045–076, 2019.

[37] “Home,” PyCaret. [Online]. Available: https://pycaret.org/.

[38] S. Ravikumar and P. Saraf, "Prediction of Stock Prices using Machine Learning (Regression, Classification) Algorithms," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, pp. 1-5, 2020.

[39] J. Tanuwijaya and S. Hansun, “LQ45 Stock Index Prediction using k-Nearest Neighbors Regression,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 2388–2391, Sep. 2019.

[40] Y. Yang and Y. Yang, "Hybrid Prediction Method for Wind Speed Combining Ensemble Empirical Mode Decomposition and Bayesian Ridge Regression," in *IEEE Access*, vol. 8, pp. 71206-71218, 2020.

[41] F. Mar'i, U. Pratiwi, I. Oktanisa and F. Utaminingrum, "Comparative Study of Numerical Methods in Multiple Linear Regression For Stock Prediction Jakarta Islamic Index (JII)," *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, Lombok, Indonesia, pp. 110-115, 2019.

[42] Q. Sun, W.-X. Zhou, and J. Fan, “Adaptive Huber Regression,” *Journal of the American Statistical Association*, vol. 115, no. 529, pp. 254–265, Apr. 2019.

[43] A. Yenkikar, M. Bali, and N. Balu, “EMP-SA: Ensemble Model based Market Prediction using Sentiment Analysis,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 6445–6452, Jul. 2019.

[44] S. S. Roy, D. Mittal, A. Basu, and A. Abraham, “Stock Market Forecasting Using LASSO Linear Regression Model,” in *Advances in Intelligent Systems and Computing*, pp. 371–381, 2015.

[45] B. M. Henrique, V. A. Sobreiro, and H. Kimura, “Stock price prediction using support vector regression on daily and up to the minute prices,” *The Journal of Finance and Data Science*, vol. 4, no. 3, pp. 183–201, Sep. 2018.

[46] C. Fraley and T. Hesterberg, “Least Angle Regression and LASSO for Large Datasets,” *Statistical Analysis and Data Mining*, vol. 1, no. 4, pp. 251–259, Mar. 2009.