IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Stock Prediction and Forecasting using Stacked LSTM-Recurrent Neural Network Model

Urmita Sharma
Department of Applied Mathematics
Delhi Technological University
Delhi, India

Sumedha Seniaray
Department of Applied Mathematics
Delhi Technological University
Delhi, India

***Abstract*— Accurately forecasting stock market returns is challenging due to the volatile and non-linear nature of financial capital markets. Artificial intelligence and increased computing power have ushered in a new era in which programmable methods of predicting market prices have proven to be more accurate. A successful prediction of a stock's future price will result in a substantial profit. In this paper, we have proposed a deep learning-based model to make prediction more reliable and simpler. The paper focuses on the use of Long Short Term Memory algorithm (LSTM), which is an advanced form of Recurrent Neural Network. We checked the accuracy of our model using stacked Long Short Term Memory and forecasted the future close prices of stock data through a backtesting method by using multi-layer LSTM networks. After performing the experiment, we were able to forecast the forthcoming 10 days closing price of the given stock.**

Keywords*— Recurrent Neural Network, Long-Short Term Memory, Deep Learning, Stock Market, Technical Analysis, Prediction, Forecasting.

I. Introduction

The stock market is one of the earliest methods for a regular citizen to exchange stocks, make investments, and profit from businesses that sell a piece of themselves on this site. As a result, techniques for forecasting stock prices in advance by analysing the trend over the previous few years have been established which may prove to be extremely useful for making stock price movements in order to maximize profit and reduce losses. In the past, two key methods for forecasting an organization's stock price were suggested.

Fundamental and Technical analysis are the two main components of stock market analysis:- The method of assessing a company's future profitability based on its current business environment and financial performance is known as fundamental analysis. Technical analysis, on the other hand, involves analysing statistical data and reading charts to assess stock market trends [1]. The emphasis of this paper is technical analysis.

As we all know, the stock market is a vital trading medium that has an effect on everyone on a personal and national level. The basic theory is straightforward: businesses would list their stock as commodities in minimal quantities known as stocks. They do it to help the organisation to raise money. The IPO, or initial public offering, is when a company sells its shares at a set price. This is the price at which a corporation sells stock in order to collect funds. The stock then becomes the owner's property, and he can sell it to anyone at any time on a stock exchange like the BSE or Bombay Stock Exchange. Traders and sellers are continuing to sell these shares for a profit at their own dime. However, the company only retains when the IPO proceeds [2].

The endless jumping of shares from one party to the next in order to make more money causes the price of a specific share to increase with each successful sale. The exchange's share price falls if the company issues more stock at a lower IPO price, and buyers and sellers lose money. In a nutshell, this occurrence is the cause of people's distrust of investing in capital markets, as well as the rise and fall in stock prices. Stock market trend prediction has also piqued the interest of statisticians and computer scientists, owing to the fact that it poses complex modelling challenges. There are methods or algorithms that can be used to forecast stock valuation with a high degree of accuracy. However, one question remains: what are the chances that an individual buying shares from a specific company would turn out to be a profitable venture or a complete failure? It could be fine to invest in a specific stock if an educated guess is made on a larger scale, taking into account the organization's current production, sales, and demand. However, expecting this to function in dynamic situations when ignoring such nuanced business concepts and variables is unrealistic [3].

In recent years, statistical machine learning has been the focus of a lot of research in this field. Various predictive models and algorithms have been proposed and tested to varying degrees of accuracy. Machine learning methods are also in their early stages of implementation. Several machine learning algorithms have recently been refined using a combination of statistics and learning models, including acritical neural networks, gradient boosted regression trees, support vector machines, and random forecast. Complex non-linear patterns as well as some associations that are difficult to detect using linear algorithms can be revealed by these algorithms. These algorithms also outperform linear regressions in terms of effectiveness and multicollinearity. This differs from conventional forecasting and diffusion approaches in a few ways. Statistical techniques such as the time series model and multivariate regression were used in early stock forecasting models [4].

Our paper primarily focuses on the use of deep learning based model which is a machine learning approach to research since it is clear that the historical data set collected is difficult to analyse without the use of data mining techniques. We took 10 years Nifty 50 monthly data (2011-2020) from Yahoo finance. For testing and predicting the movement of the stock data, we combined RNN and LSTM and proposed an RNN-based stacked-LSTM model based on multi-layer LSTM networks in this analysis. The LSTM algorithm, which is a more advanced variant of RNN, looks for the best hyperparameters for LSTM networks [13]. After checking the accuracy of our stacked LSTM model by showing low values of mean squared error (MSE) and mean absolute error (MAE) we forecasted the future 10 days closing price of our data via a backtesting method. Since we tested our LSTM model on the current dataset and predicted the future value, the forecast as a result of our proposed algorithm may help people decide whether to invest in a particular company while taking into account the chaos and volatility of the stock.

The description of sections contained in this paper is as follows: In Section II, related work on various models and machine learning algorithms for prediction is discussed. In Section III, the methodology and algorithms used in our work is discussed. In Section IV, the result is discussed. Section V deals with the conclusion and the future scope made on the basis of our study.

II. Related Work

Several studies have looked at the use of machine learning in quantitative finance, such as predicting stock prices, controlling and constricting an entire portfolio of stocks, and many other operations that can be protected by machine learning algorithms. Using machine learning in the quantitative financial sector to forecast prices has been the topic of many studies [4]. For forecasting market prices, several papers on stock prediction have been written, including linear regression, Random Walk Theory, Moving Average Convergence/Divergence, and some linear models such as Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), support vector machine (SVM), and Random Basic Function (RBF).

Kunal Pahwa, et.al. [2] proposed the use of regression and supervised learning techniques for stock prediction. They took 14 years GOOGL historical data from WIKI and used linear regression classifier (a supervised learning method). They trained the classifier to learn the pattern, find accuracy and predicted the adj. close price. V. Kranthi Sai Reddy [3] proposed a machine learning technique called support vector machine (SVM)-based train stock data and forecasted stock prices. The prediction of close prices is based on IBM Inc.'s historical data. They proposed the method for predicting the regular trend of stocks since SVM does not have the problem of overfitting. Adil Moghar and Mhamed Hamiche [4] proposed an RNN and LSTM model to forecast future stock market prices. They used yahoo finance to extract regular open price data for GOOGL and NKE on the New York Stock Exchange (NYSE). They trained the model with 80% of the data and then tested it with 20% of the data. Later, they demonstrated how much the epochs enhanced their model by demonstrating accuracy on 12;25;50;100 epochs, respectively on both stocks separately. Kai Chn, Yi Zhou, et.al. [5] proposed an LSTM approach for predicting stock returns and conducted a case study of the Chinese stock market. Their model was fitted using 900000 training sequences and evaluated using the remaining 311361 sequences. They later demonstrated that the accuracy of the LSTM model had improved. Hossein Abbasimehra, Mostafa Shabani, et.al. [6] proposed a demand forecasting approach based on a multi-layer LSTM network. The method uses the grid search method to consider different combinations of LSTM hyperparameters in order to choose the best forecasting model for a given time series. They then used demand data from a furniture manufacturer to equate their model to existing time-series forecasting techniques such as Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA). Adrian Costea [7] discussed how to evaluate the financial performance of Romanian non-banking financial institutions (NFIs) by using fuzzy logic techniques such as Fuzzy C-Means Clustering and Artificial Neural Network (ANN) techniques. The researcher used an ANN technique and genetic algorithms to find a function that converts the input performance space into a new performance class variable. Mehar Vijh, Deeksha chandola, et.al. [8] propsed the Artificial Neural Network (ANN) and Random Forest (RF) techniques to forecast the next day's closing price for five firms. They collected data of Nike, Goldman Sachs, Johnson Johnson, Pfizer, and JP Morgan Chase Co. over a 10-year period. They demonstrated how ANN outperforms RF in terms of prediction accuracy by using the metrics RMSE and MAPE and displaying their low values. Ishita Parmar, Navanshu Agarwal, et.al. [9] discussed how to predict stock prices using regression and LSTM-based machine learning techniques using data from Yahoo Finance. They used regression to reduce error and LSTM to remember data and outcomes. By using both methods, they should be able to increase prediction accuracy. However, we used a more advanced variant of the Recurrent Neural Networks (RNNs) called Stacked Long Short Term Memory (LSTM), which keeps information from previous states. RNNs (recurrent neural networks) are strong models for sequential data processing and are used to predict stock market data. The memory cell, a computational system that replaces traditional artificial neurons in the network's hidden layer, is introduced by LSTM [10]. Networks can effectively connect memories and input remote in time with these memory cells, allowing them to understand the dynamic structure of data over time with strong predictive ability. Our aim in this paper is to forecast of next 10 days based on the LSTM-RNN based model and attain the optimal prediction accuracy.

III. Methodology

A. *Understanding RNN*

Recurrent neural networks (RNNs) are the form of artificial neural network that can model sequence data. It has nodes that are linked in a graph that is powered by a time sequence. All RNNs have feedback loops in their recurrent layer. This helps them to remember information for a long time. Standard RNNs, on the other hand, can be difficult to train to solve problems involving long-term temporal dependencies. This is because the gradient of the loss function decays exponentially over time.

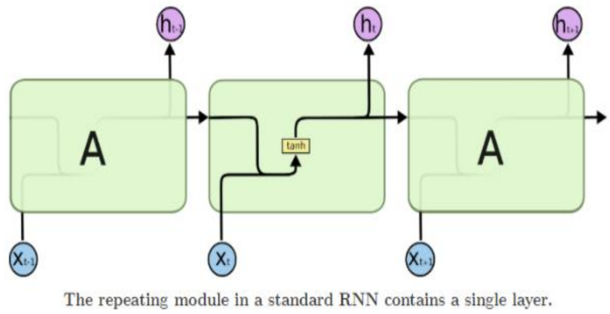

Fig. 1. The Structure of RNN with single tanh layer.

The above figure 1 shows the structure of RNN as it contains a single tanh layer which makes it difficult to store long time memory.

B. *Understanding Stacked LSTM*

Stacked LSTM architecture is basically an LSTM Model comprised of multiple LSTM layers. Long short term memory (LSTM) networks are a advance form of RNN. A 'memory cell' is used in LSTM modules that can store data for long periods of time. As information is stored in the memory, it is output, and it is forgotten, it is regulated by a collection of gates. New gates, such as input and forget gates, are implemented in LSTMs to solve the problem of loss function gradient, allowing for better gradient control and protection of "long-range dependencies." Increase the number of repeated layers in the LSTM, which we call a stacked LSTM, to solve the RNN's long-range dependence [13].

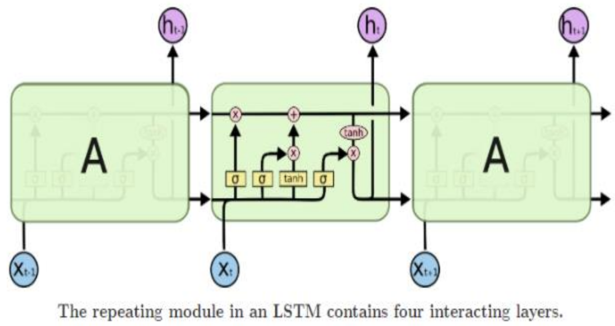

Fig. 2. The Structure of LSTM with four interacting layers.

The above figure 2 describes the composition of LSTM nodes. While LSTMs have a chain-like structure, the repeating module is different. There are four neural network layers instead of one, each interacting in a specific way. The storage of passed data streams is the responsibility of any LSTM node, which is made up of a series of cells. Each cell's upper line serves as a transport line, carrying information from the past to the present. A tanh layer is then used to build a vector of new candidate values that could be applied to the state. Cell independence aids the model's dispose filter, which adds values from one cell to the next. Finally, the sigmoidal neural network layer containing the gates moves the cell to an optimum value by disposing or allowing data to pass through [4].

C. Data Description

The project is based on 10 years monthly historical data of Nifty 50 from 2011 to 2020 which was downloaded from Yahoo Finance.

Following are the terms that are important:-

- **Open Price**- The price of a business day's first purchase.
- **Close Price**- After the trading hours of the exchange where it trades, the last price paid for a share of that stock.
- **High**- The highest price for a given time span. Low- The lowest price for a specified time span.
- **Adj. Close**- The adjusted closing price adjusts a stock's closing price to reflect its value after any business decisions have been taken into account.
- **Volume**- In the sense of a single share of stock that is traded on a stock exchange, the number of shares exchanged in a security.

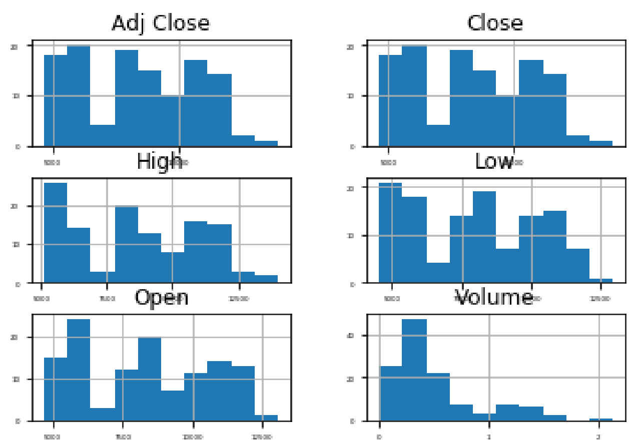

Fig. 3. Histogram of all the columns of dataframe showing the frequency of values.

Before fitting our model, we begin by analysis by calculating descriptive of the data.

	Open	High	Low	Close	Adj Close	Volume
count	120.000000	120.000000	120.000000	120.000000	120.000000	1.200000e+02
mean	8318.447901	8605.974602	8007.274183	8369.182907	8369.182907	4.643092e+06
std	2298.343767	2373.322006	2258.921885	2343.851499	2343.851499	4.227346e+06
min	4675.799805	5099.250000	4531.149902	4624.299805	4624.299805	0.000000e+00
25%	5964.800049	6134.824951	5699.025146	5923.925171	5923.925171	2.718475e+06
50%	8300.875000	8619.274903	7935.474853	8304.975097	8304.975097	3.673600e+06
75%	10450.662602	10900.550050	10098.324952	10502.312257	10502.312257	5.044475e+06
max	13062.200200	14024.849610	12962.799810	13981.750000	13981.750000	2.130300e+07

Fig. 4. Table summarizing the results of descriptive analysis of Nifty 50 Data.

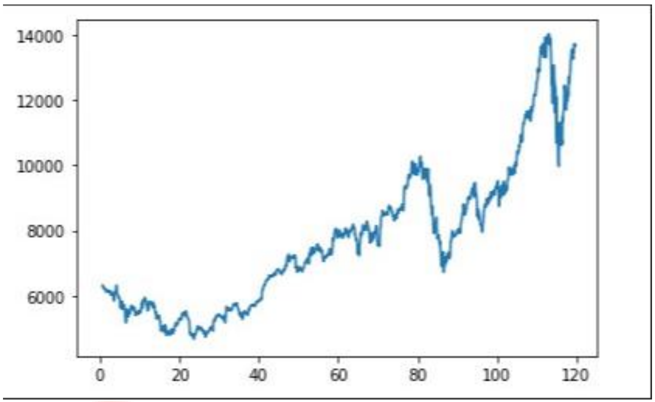

Fig. 5. Closing Price of dataframe.

Our target value is close price, since the project's main goal is to forecast future data close prices using our proposed model.

D. Data Preprocessing

Data discretization, data transformation, and data cleansing are all part of the pre-processing level. Data cleansing and data integration are two of the most important aspects of data management. To analyse, the dataset is split into two sections: training and test after it has been transformed into a clean dataset. We search for non-applicable possibilities by cleaning our files, and then we switch to feature scaling, for which we imported min-max scalar from scikit-learn, a python machine learning library. Min-Max is a technique for transforming features by scaling each one to a specific range between 0 and 1.

Splitting dataset into train and test split:- We have split our dataset in 80:20 ration in which we have trained 80% data and tested our model prediction on 20% data of the dataset. A huge amount of training data leads to a more powerful and reliable classifier, which improves overall precision. Testing is often a very simple procedure. Ascertain that your evaluation data that is at least 20% larger than or equal to our training data. Testing is a measure of the classifier's accuracy, and it's been found that testing is inversely proportional to a classifier's score on occasion.

E. Feature Extraction

In feature extraction, we have used Recurrent Neural Network (RNN). Recurrent layers are like feed forward neural networks. Only the features that will be fed into the neural network are stored in this layer are selected [15].

- *Building RNN:* For building our RNN, we have imported the Keras library and packages. We imported sequential, dense, LSTM, and dropout libraries from Keras, which is a tensor flow API for creating and draining deep learning models at a high level. The input dense layer simply represents a matrix vector multiplication, and the dense layer was used to change the proportions of our output. Sequential is a layer stack that allows one to construct a sequential model by passing a list through it.

- *Initialising RNN:* The first step in creating a deep learning model is to read the data and then allocate it to the model.
- *Building Stacked LSTM:* In this LSTM, a sequential input layer is followed by three LSTM layers: a dense layer with activation, a dense output layer with linear activation, and a dense output layer with linear activation. Dropouts are used to strengthen the neurons, enabling them to predict the pattern without relying on a single neuron.
- *Compiling the RNN:* Now, we will compile our RNN to using ADAM optimizer and loss as mean squared error which kept on decreasing when we fit our LSTM model with epochs=100, batch size=64 and verbose=1. Here, the number of epochs means how many times we go through the training set, batch size which is the hyperparameter defines the number of samples to work through before updating the internal parameters and verbose which helps to detect overfitting which happens when our model's accuracy keeps on improving.

Optimisation: Since it's nearly not possible to build a flexible classifier in a one pass, we should constantly optimise. Optimisation, in context of deep learning, is used to train the neural networks [14]. Here, we used the optimizer Adam to create the LSTM model because it has a high performance and quick convergence compared to other optimizers. We have used ADAM (Adaptive Movement Estimation) optimiser with a learning rate of 0.0005. Adam is a deep learning model training algorithm that uses stochastic gradient descent instead of stochastic gradient descent. It combines the best features of the AdaGrad and RMSProp algorithms to create an optimization algorithm for problems with sparse gradients and noisy data.

Regularization: Another important aspect of training the model is to keep the weights from being too high. As a result, there are overfits. We have chosen Tikhonov Regularization for this reason which regularised all the parameters equally.

F. Visualization

Such models are often subjected to a rolling analysis to determine their stability over time. When using a statistical model to analyse financial data, one of the most important assumptions is that the model's parameters remain constant over time. Here, we checked our model on actual close price data set and it showed promising result as the prediction of test data is correct and we predicted the output.

G. Forecasting

After predicting our test data while checking our model accuracy using train data, we afterwards forecasted the next 10 days output of our input close price. We optimised the accuracy of our model by taking into the account the values of mean squared error (MSE) and mean absolute error (MAE). The higher the model's prediction, the lower the MSE and MAE values are.

Parameter Overview:

TABLE I

LSTM MODEL SUMMARY

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 50)	10400
dropout (Dropout)	(None, 100, 50)	0
lstm_1 (LSTM)	(None, 100, 50)	20200
dropout_1 (Dropout)	(None, 100, 50)	0
lstm_2 (LSTM)	(None, 100, 50)	20200
dropout_2 (Dropout)	(None, 100, 50)	0
lstm_3 (LSTM)	(None, 100)	60400
dropout_3 (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101

Total params: 111,301
Trainable params: 111,301
Non-trainable params: 0

Table I shows the summary of stacked LSTM model used. We used 4 LSTM layers with input shape-100, hidden size-50, dropout rate-0.2 and output layer-1. For training data, we used 100 epochs, batch size 64 (a gradient descent hyperparameter that specifies how many training samples must be processed before the model's internal parameters are changed), and verbose-1 (which includes both a progress bar and one line per epoch). The number of epochs is a gradient descent hyperparameter that establishes how many complete passes through the training dataset are made. The results of our research revealed that the number of epochs as well as the length of the data have a significant impact on the testing outcome after our RNN has been trained. We saw a steady decrease in validation loss as well as the values of MSE and MAE after each epoch. The accuracy of our LSTM model was then assessed using the mean squared error (MSE) and mean absolute error (MAE) as metrics.

TABLE II

VALUES OF MSE & MAE

Epoch Size	**MSE**	**MAE**
Epoch 1	0.1588	0.3429
Epoch 100	0.0057	0.0567

Table II shows the reduction of values of MSE & MAE from epoch 1 to epoch 100. The total epoch size was 100. The value of these two metrics kept on decreasing from running epoch 1 to epoch 100. Hence, it shows that the accuracy of our model is improving.

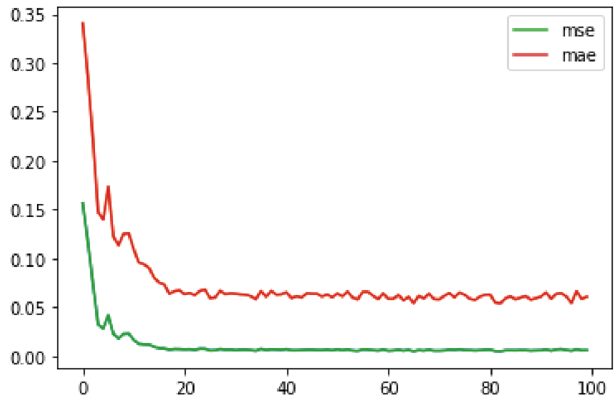

Fig. 6. Decrease in the values of MSE & MAE.

Graph in the figure 6 clearly shows the decrease in the values of MSE and MAE after training our LSTM model at each epoch. x-axis and y-axis in the graph depicts the number of epoches and value of MSE & MAE, respectively. The low values shows the accuracy and improvement of our proposed model.

IV. RESULTS

When the model is done, we use it to produce the desired results. In our case, we'll make a graph of our findings (Fig. 7) based on our criteria and requirements that we've already covered in this paper. We trained 80% data and tested on 20% data which is clearly shown in the graph below.

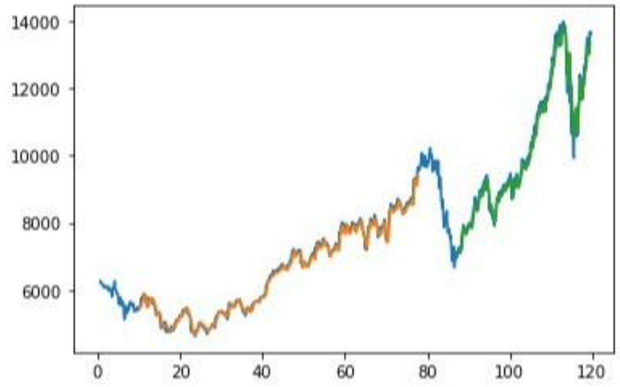

Fig. 7. Test predict of Stock price of Nifty 50 from 2011-2020 using Stacked LSTM.

In figure 7, graph shows the prediction of our model on available dataset. Blue Line represents our actual given close price data. Orange Line represents our train data of close price. Green Line represents the predicted output of test data of close price.

We have divided our data in such a way that test data will be after some specific date. The graph clearly shows that our stacked LSTM model has worked well and has performed accurately on the available dataset. Every result's most important feature is its precision. Accuracy is a component that any machine learning developers always strives to improve. Following the development of the model, an endless amount of work is expended in order to improve the model's accuracy. Our graph exactly shows the accuracy of our model.

Now, we will predict next future 10 days output. We used backtesting data to run our LSTM algorithm to forecast. We have seen our model is giving accurate results we can now rely on the model for forecasting. We have taken past 10 days data as an input from our dataset and forecasted the next future 10 days close price.

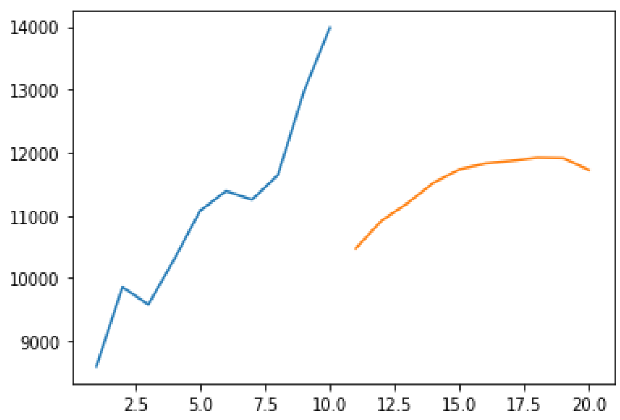

Fig. 8. The forecast of next 10 days.

In figure 8, graph shows the future prediction. Blue Line represents previous 10 days data. Orange Line represents next 10 days output. Here, x-axis depicts the input values of past data and y-axis depicts the output values we want to be foreseen.

Now, we combine the past graph with the future forecasted graph and we can see that we are getting a smoothen graph.

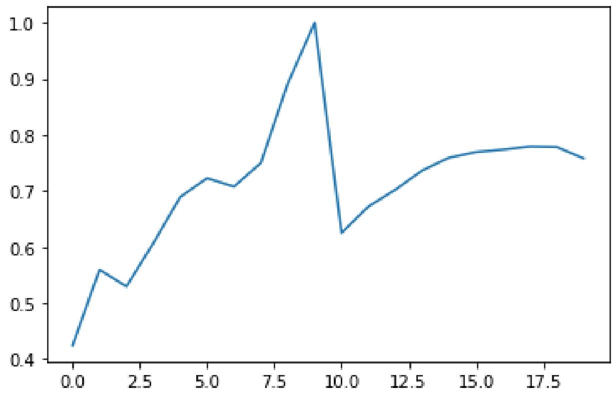

Fig. 9. Combined graph with future 10 days output prediction.

In figure 9, graph shows that the stacked LSTM model has done an incredible work of forecasting the performance for next 10 days. The implementation of this model is the easiest of them all, and it took the least amount of time, saving us time that could be spent on other important tasks.

V. CONCLUSION

This paper proposes an RNN-based stacked LSTM model for forecasting future values for Nifty 50 stock data over a 10- year period downloaded from Yahoo finance. We performed the precision of our model on the existing 80% data and tested it on 20% data, then showed the accuracy via values of MSE & MAE. We also optimised it by using ADAM optimiser and forecasted the market data's next 10 days future closing price through backtesting method since our target value is close price and our model is providing promising result. The main goal here is to find the most accurate qualified algorithm for predicting potential values. The experimental results depict that our model has yielded some positive results. The results of the testing showed that our model can monitor the evolution of near prices using the backtesting process. This approach has shown an increase in prediction accuracy, resulting in positive outcomes. We showed that our proposed model worked accurately on available dataset which further helped in predicting the forthcoming 10 days close price of our current dataset. We used a deep learning approach to assess this precision, but most importantly, we focused on the backtesting method for future close price prediction. In future, the accuracy of this model can be improved by other methods too like using bi-directional LSTM and optimising a hyperparameter to combat overfitting when backtesting the data which can improve the further accuracy of the model. In addition, other machine learning models could be investigated to see what accuracy rate they produce.

References

[1] Ashish Pathak, Nisha P Shetty, "Indian Stock Market Prediction using Machine Learning and Sentiment Analysis", December 2017.

[2] Kunal Pahwa, Neha Agarwal, "Stock Market Analysis using Supervised Machine Learning", 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, Feb 2019.

[3] V Kranthi Sai Reddy, "Stock Market Prediction Using Machine Learning", International Research Journal of Engineering and Technology (IRJET), Oct 2018 .

[4] Adil MOGHAR, Mhamed HAMICHE, "Stock Market Prediction Using LSTM Recurrent Neural Network", International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI 2020) April 6- 9,2020, Warsaw, Poland.

[5] Kai Chen, Yi Zhou, Fangyan Dai, "A LSTM-based method for stock returns prediction : A case study of China stock market", 2015 IEEE International Conference on Big Data (Big Data).

[6] Hossein Abbasimehra, Mostafa Shabanib, Mohsen Yousefic, "An optimized model using LSTM network for demand forecasting", Computers Industrial Engineering 143 (2020) 106435.

[7] Adrian Costea, "Applying fuzzy logic and machine learning techniques in financial performance predictions", 7th International Conference on Applied Statistics.

[8] Mehar Vijha, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, "Stock Closing Price Prediction using Machine Learning Techniques", International Conference on Computational Intelligence and Data Science (ICCIDS 2019).

[9] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan, "Stock Market Prediction Using Machine Learning", December 2018.

[10] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, Xiao-Yang Liu, "DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News". 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[11] Shubharthi Dey, Yash Kumar, Snehanshu Saha, Suryoday Basak, "Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting", October 2016.

[12] Ha Daewoo, Kim Youngmin, Jaejoon Ahn, "A Study on the Prediction of the Rise and Fall of the KOSPI 200 Stock Index Using the XGBoost Model", Journal of the Korean Data Information Science Society, vol.30(3), pp. 655-669,2019.

[13] Gers, Felix A., Nicol N. Schraudolph, Jurgen Schmidhuber, "Learning precise timing with LSTM recurrent networks", Journal of Machine Learning Research, 2002.

[14] Abadi, Martin,"Deep learning with differential privacy", Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016.

[15] Hiransha M , Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P, "NSE Stock Market Prediction Using Deep-Learning Models", International Conference on Computational Intelligence and Data Science (ICCIDS 2018).