Journal of Informatics
Electrical and Electronics Engineering, 2022,
Vol. 03, Iss. 01, S. No. 008, pp. 1-13
ISSN (Online): 2582-7006

Stock Price Prediction using ML and LSTM based Deep Learning models

Faraz Ahmad[1], Pawan Singh[2]

[1,2]Department of Computer Science Engineering, Amity School of Engineering and Technology, Amity University Uttar Pradesh, Lucknow Campus, India
[1]farazahmad2581@gmail.com, [2]pawansingh51279@gmail.com

How to cite this paper: F. Ahmad, P. Singh, "Stock Price Prediction using ML and LSTM based Deep Learning models," *Journal of Informatics Electrical and Electronics Engineering* (*JIEEE*), Vol. 03, Iss. 01, S No. 008, pp. 1–13, 2022.

http://doi.org/10.54060/JIEEE/003.01.008

Received: 05/04/2022
Accepted: 24/04/2022
Published: 25/04/2022

Abstract

Stock Price Prediction has become an important area of research for such a very long time. A lot of research has already been made to predict the stock in a best possible manner and to gain more profit from that , Now adays some market hypothesis believe that it is nearly very difficult to predict the stock price accurately but at the same time some machine learning techniques proved that choosing of right model and appropriate variables may lead to scenario where stock prices and their movement can be easily predicted with great accuracy. Prediction of stock price becomes easy due to the introduction of data mining techniques which helps the researchers to identify meaningful patterns and find the best possible results by working on the technical analysis of stock. In this research we have implemented some of the machine learning and deep learning techniques to gain more insights of varying stock prices with respect to time the purpose of introducing the Deep Learning model is that they can predict more accurate results as they are the advanced version of Machine Learning models. We have also compared these Machine Learning and Deep Learning models so that we can get the best possible model for our project.

Keywords

Stock Price Prediction, Regression, Long Short-Term Memory, Multivariate Time Series, Linear Regression, Support Vector Machine, Random Forest, walk Forward Validation.
.

1. Introduction

The main objective of our research is to develop a framework such that by the help of various models we can easily predict the stock price movement and that with a fairly great accuracy. This research includes the use of various Machine Learning and Deep Learning models. Nowadays, companies are using such machine learning techniques to meet consumer expectations. There are two approaches for prediction using machine learning techniques: supervised and unsupervised, which comes under artificial intelligence and machine learning. The critical difference between these two is that one uses labelled data, and the other uses supervised unlabeled data, as the name suggests. Moreover, unsupervised is done without any supervision. Supervised learning algorithms belong to the classification category of machine learning, and unsupervised learning comes under the regression category.

The thing people should keep in mind while using machine learning is the dataset they are using. It should be as solid as possible with slight variations or fluctuation. The next technique is by prediction with deep learning models and for such case we have used LSTM based deep Learning models so that they can predict and gives us good and accurate results. A lot of research has already been made to predict the stock in a best possible manner and to gain more profit from that , Now adays some market hypothesis believe that it is nearly very difficult to predict the stock price accurately but at the same time some machine learning techniques proved that choosing of right model and appropriate variables may lead to scenario where stock prices and their movement can be easily predicted with great accuracy.

In this research we have tried to build a robust framework for stock prediction by identifying the meaningful patterns from the dataset and by combining the text mining and natural language in various machine learning techniques like regression for unsupervised and classification for supervised techniques. We have also come to know that we have got a high level of accuracy when predicted the NIFTY50 index values and also same as that for various CNN models.

2. Problem Statement

K- Our goal is to collect NIFTY50 stock prices from India`s NSE for five and a half years and create a comprehensive forecasting framework for predicting NIFTY50 index values. I think so. Machine learning or deep learning models can theoretically be learned from the properties of the dataset. Daily NIFTY50 index value past movement patterns and these functions can be used to reliably estimate future index values.

NIFTY 50 is a sequence of 10 numbers. In this proposal, we chose as the forecast period for machine learning models, 1 week for deep learning model. The goal of our project is to collect the stock price for certain organization which is varying for over a long period of time we have collected the data for stock prices varying for 6 to seven days and performed basic algorithm of Machine Learning and Deep Learning to understand the meaningful patterns inside the data and how the stock prices are varying and what can be the future predictions. We come to know that it is possible for Machine Learning and Deep Learning models to analyze the past data and make accurate predictions on the future values. We have choosed a horizon of about nearly 1 year for machine learning models and for deep learning we had choosed 1 week and came to know that these values can be predicted with a very good accuracy. We have used LSTM based model so it can be better for medium term investors who are interested in putting their money and effort for short time like weekly forecast of the stock.

3. Related Work

The existing model on the forecasting of stock price is broadly divided into two categories of machine learning supervised and unsupervised and three clusters which is based on the value of the variables and the attempt or approach to solve any existing problems. The first category of our work shows the models which are implemented on cross sectional data but there are some flaws in those models as these models are very simple and basic type of Machine Learning model so they failed to produce nearly accurate results most of the time. The second category include various powerful models like (ARIMA and VAR etc) and the third category of our project include the learning method approaches using the Deep Learning and Machine Learning techniques.

AS we all know Machine Learning and Deep Learning has the power to predict the data with a very good accuracy so we have tried to implement few models to predict our data and to achieve good results. In this project I have tried to implement the LSTM (long short-term memory), and network based deep Learning models and tried to find the best possible results using the past knowledge of stock prices.

4. Methodology

4.1. Data Pre-Processing

In the above sections I have mentioned that in this research we have to collect the data for stock prices and our aim is to build the predictive framework for forecasting the daily price variation of nifty50 companies. I have collected some of the company's data from the website named Kaggle and store it in my database. The data was collected from December 14 2017, till July 15 2018 from some of the finance websites the data is in the raw format which was having not accurate values so cleaning of data was also necessary. We have imported the data from our database and performed all the process of data cleaning to make the data ready to use, some of the techniques that is used for cleaning process include

a.Data Cleaning

- Handling of missing values
- Ignoring the tuples/Removing the tuples
- Filling the missing values
- By median
- Handling the noisy data by using Binning or soothing method

b. Data Integration

In this process the heterogenous is combined from multiple sources into one common source

c. Data Reduction

- Data cube aggregation
- Dimension Reduction
- Numerosity Reduction
- Attribute subset Selection

d.Data Transformation

- Normalization- Min max Normalization
- Z score Normalization and
- Decimal Normalization.

The Raw data of nifty50 values consist of the following variables

i. Data
ii. open value
iii. close value
iv. High value
v. low value
vi. volume of the index

4.2. Performing the Required Operations

Firstly, we have followed the regression approach and take the open value as the primary and the responsible value and all the other values as the dependent values. We have carried some preprocessing of the data which I have already mentioned above the techniques that we have used for cleaning the data and making the data free from any faulty values. The following 5 variables are used in our project that includes High_norm, close_norm, volume_norm, range_norm, low_norm. After importing the data and performing all the preprocessing techniques on to the nifty50 data for the period of Dec 14 to July 15 the processed data is further used for building and testing the regressions models of Machine Learning and Deep Learning.

The data is then divided into training and testing part in the ratio of 75% and 25 % for training we used the data for December 12, 2017(which was Tuesday) till December 19(which was Saturday). The models are then implemented on the training data to predict the future outcomes. The training dataset consist of 2095 records that included weeks of about 319. In Machine Learning models implementations, we used the daily data from the training dataset and predict the open value for everyday in the testing phase and for the deep learning-based models a different approach has been followed which is basically known as walk forward validation approach or multi-step forecasting approach.

In order to build our framework more robust we have tried some of the deep learning model on to the dataset firstly we have tried to implement the CNN model and demonstrated its efficiency and performance secondly, we have used LSTM (Long Short-Term Model) for our prediction. For this project we are using a live dataset of Stocks of Microsoft Corporation (MSFT) form tiingo.com. To begin with we first got the API key in the form of a token number from tiingo.com. Then we read the dataset using pandas

```
In [4]: import pandas_datareader as pdr
        import datetime
```

```
In [5]: dt=pdr.DataReader('msft','tiingo','2000-1-1',datetime.datetime.now(),api_key='911ee28d70118f9cea5a84d2b8f1436fa32d3116')
```

Figure 1: Importing the dataset

DATA_READER function. After viewing the data set using head function, we analyzed the data and learned that only the columns adjOpen, adjClose, adjHigh, adjLow, adjVolume will serve as a good input and output. Among these columns we will use adjOpen, adjHigh, adjLow, adjVolume as a feature set(X). In the feature set we will add a new column by the name new_Close which we will predict (target value akaY). We are doing so as we do not want to lose the adjClose values as it will be beneficial for us in plotting the graph.

```
In [6]: dt.head()
Out[6]:
```

symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
	2000-01-03 00:00:00+00:00	116.56	118.62	112.00	117.37	26614200	37.281068	37.939947	35.822577	37.540142	53228400	0.0	1.0
	2000-01-04 00:00:00+00:00	112.62	117.12	112.25	113.56	27059500	36.020881	37.460181	35.902538	36.321535	54119000	0.0	1.0
msft	2000-01-05 00:00:00+00:00	113.81	116.37	109.37	111.12	32029800	36.401496	37.220297	34.981386	35.541114	64059600	0.0	1.0
	2000-01-06 00:00:00+00:00	110.00	113.87	108.37	112.19	27488300	35.182888	36.420686	34.661542	35.883348	54976600	0.0	1.0
	2000-01-07 00:00:00+00:00	111.44	112.25	107.31	108.62	31006800	35.643464	35.902538	34.322507	34.741503	62013600	0.0	1.0

Figure 1.1: raw dataset

Now we will copy the values of adj Close to new Close and performance shift operation to it. We will now split the feature set(X) into two parts X1 and X2 such that X1 is having values from start to except last 5 and X2 will be having last 5 values. Here 5 is just an example, it is the number of days data that is to be predicted. X1 and Y is used for training the model and X2 is used for predicting.

```
In [16]: dt['close'].plot()
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x44d38ef7c8>
```

Figure 1.2: graph plotting

We will now split X1 and target value into two types of dataset namely- training data set(x1_tr, y_tr) and the testing data set(x1_ts ,y_ts). We do it by using the train_test_split function from the sklearn model selection library. Now using the fit function under the linear regression model we will fit the training data in the model. And predict the new_close values using the predict function of linear regression model. The predict function will take X2 as a parameter to give the prediction. We can also find the efficiency of our model by using the score function of Linear Regression model which takes X1_ts and Y_ts as its parameter.

```
In [24]: from sklearn.model_selection import train_test_split
         from sklearn.metrics import mean_squared_error

In [25]: x_tr,x_ts,y_tr,y_ts=train_test_split(x1,y,test_size=0.25)

In [26]: from sklearn.linear_model import LinearRegression
         algo=LinearRegression()
         algo.fit(x_tr,y_tr)
         algo.score(x_ts,y_ts),mean_squared_error(y_ts,algo.predict(x_ts))

Out[26]: (0.99667992136321, 4.6269766786197355)
```

Figure 1.3: testing the data

4.3. Performing some LSMT based deep learning models

We also build deep learning-based regression models to make the prediction framework more robust and accurate. In one of his previous papers, he showed the effectiveness and effectiveness of convolutional neural networks (CNNs) in predicting time series index values [28]. In this task, we used the predictive power of another deep learning model. A long short-term memory (LSTM) network for predicting complex multivariate time series such as the NIFTY50 series. LSTM is a special type of recurrent neural network (RNN). A neural network that allows a feedback loop to send data from a node in the forward layer to a node in the reverse layer [26]. In an RNN network, the output of the network in a particular time slot depends on the state of the network in the previous time slot, as well as the input to the network in a particular time slot. However, RNNs have a problem called the vanishing gradient problem. In this problem, the network either stops learning or continues learning at a very high learning rate, so it does not converge to the point of minimum error [26]. LSTM networks overcome the vanishing gradient problem by intelligently forgetting irrelevant information from the past. Therefore, we find that such networks are well suited for modeling sequential data such as text and time series. An LSTM network consists of memory cells that use memory to maintain state information over time and a gating unit that regulates and controls the flow of information through them. The LSTM network uses three types of gates: oblivion gates, input gates, and output gates.

The Gate of Oblivion helps you to throw away irrelevant past information and remember only the information related to the current slot. The input gate controls new information that acts as an input to the current state of the network. Old information from the oblivion gate and new information from the input gate are effectively aggregated by the cell state vector. Finally, the output gate produces output from the network in the current slot. This output can be thought of as the predicted value calculated by the model for the current slot. The architecture of LSTM networks integrated with the BPTT (Backpropagation Through Time) algorithm for learning parameters provides these networks with high levels of performance in univariate and multivariate time series predictions.

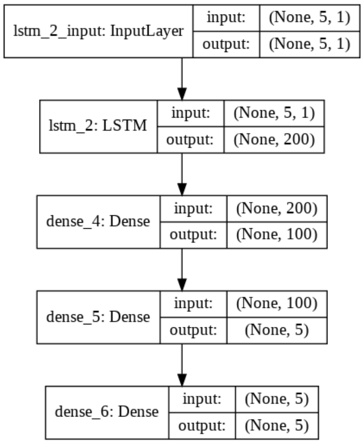

Fig 2: The architecture of univariate LSTM model with prior one week's data as the input

Leverage the power of LSTM models in multi-level time series predictions using walk-forward validation methods [27]. With this technique, you need to make a one-week forecast on the model and then use the actual data for that week in the model to make the next week's forecast. This is realistic and practical because most real-world applications do not use a prediction period longer than a week.

This task used four different LSTM models. This approach depends on the model's architecture and the format of the input data that the model uses. The four models are: (I) Multi-level predictive LSTM model with 1-week univariate input data, (ii) Multi-level predictive LSTM model with 2-week univariate input data, (iii) Multi-level for prediction with Coder-DecoderLSTM2 Weekly Coder-Decoder-LSTM Multilevel prediction using univariate input data and bivariate data. The architectural design and parameters for each of the four models are described below.

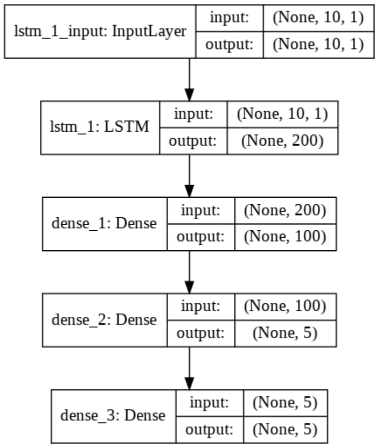

Figure 2.1. The architecture of univariate LSTM model with prior two week's data as the input

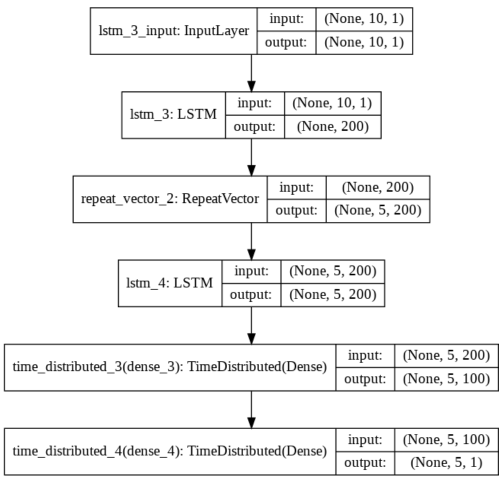

Figure 2.2. The architecture of univariate encoder-decoder LSTM model with prior two weeks' dataas the input

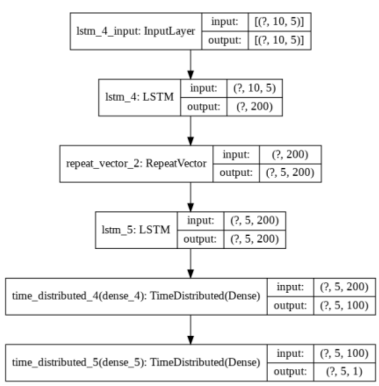

Figure 2.3. The architecture of multivariate encoder-decoder LSTM model with prior two week's data as the input

5. Performance of Machine Learning models

As mentioned earlier that we have implemented machine Learning models first and the results of several models as shown above. For this evaluation of machine Learning models two metrics has been used for the same. These metrics are 1. Product moment correlation and 2. Ratio of root mean square and RMSE value. After implementing all the machine Learning models we come to know that these MARS, RF and Multivariate Regression has outperformed of all the other models which is basically predicting the values based on the actual value.

Table 1. Boosting regression results

Stock	Case I Training Data		Case II Test Data	
NIFTY 50	Correla-tion RMSE	0.99 0.37	Correla-tion RMSE	0.98 1.87

Table 2. Random forest regression results

Stock	Case I Training Data		Case II Test Data	
NIFTY 50	Correla-tion RMSE	0.99 0.29	Correla-tion RMSE	0.99 0.42

Table 3. ANN regression results

Stock	Case I Training Data		Case II Test Data	
NIFTY 50	Correla-tion RMSE	0.67 12.77	Correla-tion RMSE	0.44 19.31

Table 4. SVM regression results

Stock	Case I Training Data		Case II Test Data	
NIFTY 50	Correla-tion RMSE	0.99 0.75	Correla-tion RMSE	0.58 8.40

These RF and multivariate model give lowest ration of RMSE to the mean of the actual values and hence we can say that these two models are the best models of Machine Learning for which can be used for our predictions. The performance of all models in training and testing Multivariate regression, MARS, and Random Forest are more important than all other models in terms of the metric correlation coefficient between the actual predictive starting values of the test dataset because of the importance of test performance. is. It's important. You can see that it's great. However, the lowest ratio of RMSE to the average of the actual open values is the result of multivariate regression and random forest. Therefore, based on the performance of all machine learning models, we conclude that multivariate regression and random forest regression are the most accurate models in terms of predictive accuracy of NIFTY 50 time series.

Artificial Neural Network

STOCK	CASE I- TRAINING DATA		CASE II- TEST DATA	
Dataset	Correlation RMSE	0.67 12.77	Correlation RMSE	0.44 19.31

Figure 3. Artificial Neural Network

STOCK	CASE I- TRAINING DATA		CASE II- TEST DATA	
Dataset	Correlation RMSE	0.99 0.75	Correlation RMSE	0.58 8.40

Figure 3.1. Support vector Machine

6. Performance of Deep Learning models

So, we now focus on performance of some of the deep learning LSTM based models. The performance of each of the lstm model is calculated in such a way that we implemented all these models on to the testing data for over 10 rounds.

On implementing the LSTM based deep Learning models on to the dataset we aim to know that the nifty50 dataset does not inhibit multivariate characteristics and we came to the conclusion that the univariate LSTM model which are used for prediction of 1 week data given one week's prior data comes out to be more accurate as compared to the other LSTM models. The table shown below depicts the performance of LSTM 1 model

Table 5. LSTM regression results – univariate time series with previous week data as the training input (LSTM#1)

No.	RMSE	Mon	Tue	Wed	Thu	Fri	Time
1	350.7	250	295	343	396	437	19.14
2	347.2	232	303	341	390	435	16.42
3	351.9	243	296	349	398	439	19.80
4	323.6	210	273	305	369	419	19.36
5	37.4	253	285	336	388	441	18.52
6	314.5	201	259	299	359	411	18.54
7	330.8	234	276	322	369	419	18.85
8	340.1	228	278	326	393	434	18.55
9	378.1	251	385	341	412	467	18.87
10	361.5	219	284	338	450	456	18.35
Mean	344.57	232	293	330	392	436	18.64
Min	314.5	201	259	299	359	411	16.42
Max	378.1	253	385	349	450	467	19.80
SD	18.47	17.9	34.5	16.7	25.8	17.0	0.899
RMSE/Mean	0.0311	0.02	0.03	0.03	0.04	0.04	

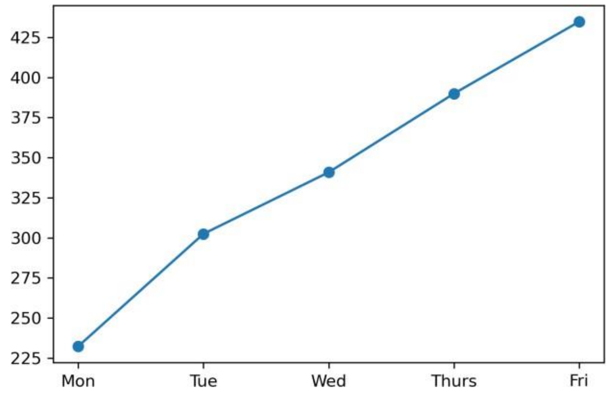

Figure 3.2 Day-wise RMSE of LSTM#1 – univariate time series with one-week data as the input

Table 6. LSTM regression results – univariate time series with the previous two weeks' data as the training input (LSTM#2)

No.	RMSE	Mon	Tue	Wed	Thu	Fri	Time
1	393.1	336	363	390	439	429	31.29
2	369.4	293	343	373	413	411	31.40
3	368.0	318	346	363	410	396	31.36
4	431.9	367	409	398	528	440	31.95
5	397.9	343	383	376	452	425	31.49
6	391.6	318	360	397	439	429	31.45
7	408.2	356	414	397	448	421	31.44
8	363.9	304	337	357	406	405	31.53
9	395.1	345	369	404	438	413	31.03
10	385.5	322	353	389	435	418	31.44
Mean	390.46	330	367	385	441	419	31.44
Min	363.9	293	337	357	406	396	31.03
Max	432	367	414	404	528	440	31.95
SD	20.5	23.2	26.6	16.4	34.7	12.9	0.23
RMSE/Mean	0.0353	0.03	0.03	0.03	0.04	0.04	

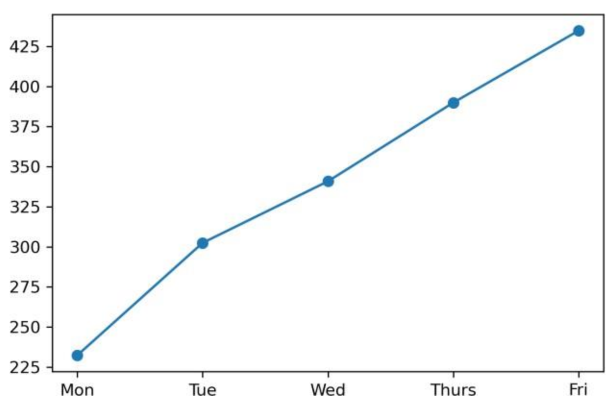

Figure 3.2. Day-wise RMSE of LSTM#1 – univariate time series with one-week data as the input

Table 6 shows the performance of the LSTM # 2 deep learning regression model. The average execution time for the 10 rounds of the model in the same computing environment was 31.44 seconds. This was almost twice as long as it took to run the LSTM # 1 model. The average is a ratio of the average RMSE of the actual modeled aperture values was 0.0353, while the average RMSE was 390.46. Therefore, the LSTM # 2 model is inferior to the LSTM # 1 model in both RMSE metrics. That means open value and average lifetime.

Table 7. LSTM regression results – univariate time series with the previous two weeks' data as the training input (LSTM#2)

No.	RMSE	Mon	Tue	Wed	Thu	Fri	Time
1	391.7	318	383	395	433	418	12.79
2	418.1	367	398	415	459	446	12.56
3	409.1	334	381	403	462	452	14.95
4	423.0	365	400	413	467	461	14.74
5	403.4	326	414	389	424	453	14.79
6	397.9	349	379	393	440	422	14.68
7	389.8	344	384	372	425	418	15.11
8	395.6	327	362	391	445	440	15.44
9	449.0	343	387	468	527	493	14.95
10	412.1	348	382	411	456	453	15.26
Mean	408.97	342	387	405	454	446	14.53
Min	389.8	318	362	372	424	418	12.56
Max	449.0	367	414	468	527	493	15.44
SD	17.92	16.2	14.0	25.7	29.3	22.9	1.00
RMSE/Mean	0.0369	0.03	0.04	0.04	0.04	0.04	

From the above figure we concluded that unlike the LSTM1 model the LSTM 2 model exhibit different behaviors, the RMSE value for LSTM 2 model increased consistently from Tuesday till Saturday. Below table shows the results of the LSTM 3 model implemented on to the training dataset [8-10].

Table 7. Encoder decoder LSTM regression results – univariate time series with previous twoweeks' data as the training input (LSTM#3)

No.	RMSE	Mon	Tue	Wed	Thu	Fri	Time
1	391.7	318	383	395	433	418	12.79
2	418.1	367	398	415	459	446	12.56
3	409.1	334	381	403	462	452	14.95
4	423.0	365	400	413	467	461	14.74
5	403.4	326	414	389	424	453	14.79
6	397.9	349	379	393	440	422	14.68
7	389.8	344	384	372	425	418	15.11
8	395.6	327	362	391	445	440	15.44
9	449.0	343	387	468	527	493	14.95
10	412.1	348	382	411	456	453	15.26
Mean	408.97	342	387	405	454	446	14.53

Min	389.8	318	362	372	424	418	12.56
Max	449.0	367	414	468	527	493	15.44
SD	17.92	16.2	14.0	25.7	29.3	22.9	1.00
RMSE/Mean	0.0369	0.03	0.04	0.04	0.04	0.04	

Now the Last model LSTM 4 model seems not good as it takes more time that the other three models it is very heavy model for prediction and it takes almost 5 times more time than other models and it is also depicted that this model gives a very high value for the ration or RMSE and the mean open value

Table 8. Encoder decoder LSTM regression results – multivariate time series with previoustwo weeks' data as the training input (LSTM#3)

No.	RMSE	Mon	Tue	Wed	Thu	Fri	Time
1	1329.8	1396	1165	949	1376	1655	72.79
2	2798.8	3253	3131	2712	2407	2369	69.01
3	2764.2	2588	2926	2761	3100	2391	79.88
4	1754.2	1402	1856	1766	1543	2114	62.76
5	1217.4	1521	1098	1045	1001	1340	59.61
6	1162.9	1421	1100	1075	920	1239	72.28
7	2485.4	2034	2108	2258	2767	3091	68.76
8	1788.6	1280	1590	1705	1962	2252	62.82
9	1451.6	1921	1317	1367	1362	1179	62.42
10	2185.6	1191	1373	1901	2601	3194	58.73
Mean	1893.85	1801	1766	1754	1904	2082	66.91
Min	1162.9	1191	1098	949	920	1179	58.73
Max	2798.8	3253	3131	2761	3100	3194	79.88
SD	1236.93	1527	1209	1351	1140	1032	0.624
RMSE/Mean	0.1711	0.16	0.16	0.16	0.17	0.19	

6. Conclusion

In this research paper, we have implemented several Machine Learning and Deep Learning models for the prediction of stock price over a period of time, several analyses have been performed during this research the models' accuracies have been compared which helps in choosing the right model for future predictions. The machine learning models that are used in this research p are Linear Regression, support vector machine, random forest etc. and the deep learning models include all the different types of LSTM model and CNN model. The dataset that is used in this project is NIFTY50 and the predictions are done on to the same dataset all the models are implemented and the best possible outcomes are generated.

Data Preprocessing and other several techniques has been carried out on to the data in order to get the desired results of prediction, after preprocessing the dataset gets divided into training and testing part all the models get trained on to the training data and then they are checked over testing data. After implementing all the models from both the branches of AI i.e; ML and deep learning we came to the conclusion that among all the models the LSTM based models which comes under the category of Deep Learning are found to be more superior, faster and accurate then all the other models. By studying all these models, it is concluded that the deep learning models have much higher capability for predicting the accurate results as compared to the machine learning models. It also shows that the multivariate LSTM models are not that good and accurate as compared to the univariate LSTM based regression models which are faster and more accurate.

Reference

[1]. M. S. Yang, S. J. Chang-Chien and Y. Nataliani, “A fully-unsupervised possibilistic c-means clustering method,” IEEE Access, vol. 6, pp. 78308–78320, 2018.

[2]. J. MacQueen, “Some methods for classification and analysis of multivariate observations,” Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281-297, University of California Press, 1967.

[3]. D. N. Geary, G. J. McLachlan, and K. E. Basford, “Mixture models: Inference and applications to clustering,” J. R. Stat. Soc. Ser. A Stat. Soc., vol. 152, no. 1, p. 126, 1989.

[4]. A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm”, Journal of the Royal Statistical Society, Vol. 39, Iss. 1, pp. 1-38, 1977.

[5]. Z. Lv, T. Liu, C. Shi, J. A. Benediktsson and H. Du, "Novel Land Cover Change Detection Method Based on k-Means Clustering and Adaptive Majority Voting Using Bitemporal Remote Sensing Images," in IEEE Access, vol. 7, pp. 34425-34437, 2019, doi: 10.1109/ACCESS.2019.2892648.

[6]. E. Feyen, J. Frost, L. Gambacorta, H. Natarajan, and M. Saal, “Fintech and the digital transformation of financial services: implications for market structure and public policy,” *Bis.org*, 2021. [Online]. Available: https://www.bis.org/publ/bppdf/bispap117.pdf. [Accessed: 27-Jul-2022].

[7]. Y. Meng, J. Liang, F. Cao, and Y. He, “A new distance with derivative information for functional k-means clustering algorithm,” Inf. Sci. (Ny), vol. 463–464, pp. 166–185, 2018.

[8]. M. D. Buhmann *et al.*, “Recommender Systems,” in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 829–838.

[9]. A. Singh, P. Singh, “Image Classification: A Survey. Journal of Informatics Electrical and Electronics Engineering”, vol. 01, Iss. 02, S. no. 2, pp. 1-9, 2020.

[10]. N. Srivastava, U. Kumar and P. Singh, “Software and Performance Testing Tools. Journal of Informatics Electrical and Electronics Engineering”, Vol. 02, Iss. 01, S. No. 001, pp. 1-12, 2021.