Stock Predictions using LSTM

Anshul Sharma
Masters of Applied Computing
University of Windsor
Student ID:110016388
sharm192@uwindsor.ca

Guntas Kaur Grewal
Masters of Applied Computing
University of Windsor
Student ID: 110009073
grewa12w@uwindsor.ca

Lovejot Singh Pannu
Masters of Applied Computing
University of Windsor
Student ID: 110013573
pannul@uwindsor.ca

***Abstract*—Stock markets have always been a hot spot for the investors as it gives a high return, and it changes continuously. As a result, the prediction of stock prices becomes necessary. It is a research topic for a long time, but the results are not very accurate as stock prediction includes uncertainties and the involvement of a large number of variables in a very dynamic and complex environment makes it even more convoluted. There has been a lot of research on this sequential prediction problem using Machine learning models, and recent results have shown that among all these models, LSTMs give the most effective solution [1]. LSTM is different from primitive feed forwarding RNN models as it can remember specific patterns for a long duration. In this paper, we used LSTM to predict the future price trends of various stocks using the price history of that stock and other technical aspects affecting the price. We have applied one prediction model and determined its efficacy through the final results. Finally, we were able to achieve 73.6 percent accuracy in predicting a particular stock price in the next few hours.**

***Index Terms*—Machine Learning, Stock Market Predictions, Long Short Term Memory (LSTM), Recurrent Neural Networks**

I. Introduction

All the companies in the world have their equity in the market known as stock. Every stock has some value called stock price. In today's era, many people are investing in the stock market with the hope to make significant profits. The accurate prediction of a stock's future trend will lead to considerable profits for the investors and traders. Predicting stock prices can play a vital role here. Therefore, the stock market is the major attraction of all the data scientists and research scholars for the last two decades. There have been many studies related to this sector, and most of these research studies are to predict stock price trends. Nevertheless, the results were not that efficient. These problems are termed as Sequence prediction problems, and considering data science history, these are considered the most challenging problems to solve for data scientists because of their dynamic and complex nature. Stock price prediction has a vast number of variables and uncertain depending factors, making it very difficult to predict. With the evolution of Machine learning and artificial neural networks, future stock prices can now be predicted based on the history of price data and some other technical variables. Recurrent neural networks (RNN) have been used in the past for this purpose. It is a type of artificial neural network based on feed forwarding that has an internal memory that can be used to process different sequences of inputs. This project has used Long-Short-Term Memory (LSTM) Recurrent Neural Network for stock price predictions. We have used python libraries to grab the stock market's price history data of various stocks and implemented the LSTM model to predict the future price of a particular stock. LSTMs are a special kind of RNN, capable of learning long-term dependencies. RNN can be very useful in predictions, but they fail in learning long-term dependencies. It becomes more difficult for RNN to make the required prediction as the dependency gap increases. LSTMs are programmed to overcome this problem. They can remember information for an extended period because of its unique storage unit structure, depicted in Figure 2. All RNNs have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module has just a single layer[2].

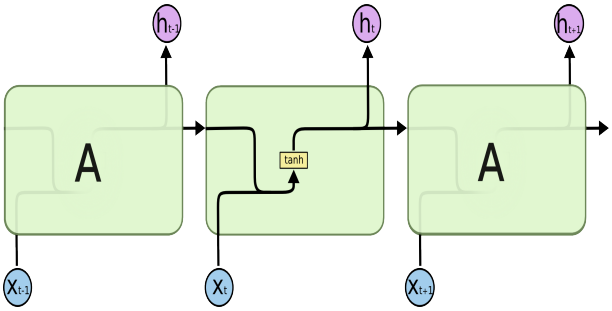

Figure 1- Repeating module in standard RNN

On the other hand, LSTMs repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting extraordinarily.

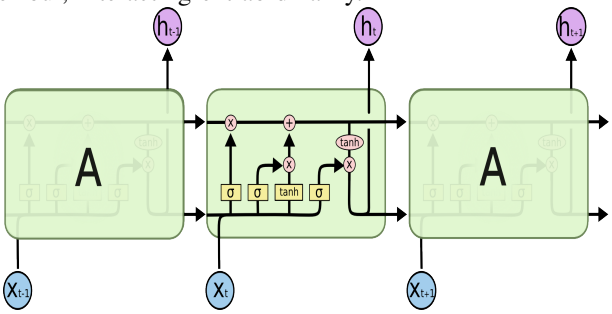

Figure 2- Repeating module in LSTM

The dataset used in this project consists of stock related data from Yahoo Finance and it has been incorporated using an API. This data is pre-processed and fed to the LSTM model. The model will be trained, tested, and then try to make predictions of a particular stock's stock prices. Finally, this paper is organized in the following manner, and Section 2 explains the problem statement to understand the purpose of performing stock market analysis. Section 3 lists and briefly describes the research papers and articles that present varying methodologies of performing predictions. Section 4 provides comprehensive knowledge about the methods and algorithms implemented to complete the project work. Further, Section 5 delineates the practical aspects of the experimentation, and it also includes discussions about the experimental datasets, workflow and the derived solutions. In the end, Section 6 suggests potential improvements and extensions to the project followed by a definitive conclusion.

II. PROBLEM STATEMENT

Predicting accurate stock market prices and trends has been an area of interest for machine learning enthusiasts over the last few years. The main aim of stock market price predictions is to drive future investments in the correct direction. Still, many factors like rates of interest, global phenomenon, political implications and the volatile economic growth rate may result in inaccuracies and unnecessary data while making predictions. It can prove to be an arduous task due to its inherent intricacy due to multiple financial indicators and the dynamic nature of the stock market trends, resulting in uncertainty. Hence, the use of apropos prediction models and techniques are required to retrieve the most prevalent information. In general, there are two ways which are used to make predictions: either (1) using the previously available data to produce future results, which is known as technical analysis, or (2) analyzing unstructured textual information to get the required results, and this method is called fundamental analysis. For the project and this report, technical analysis using machine learning principles and neural networks have been chosen. To elaborate, the ever-changing stock market prices can be predicted to a certain considerable level of accuracy using machine learning algorithms as these algorithms are designed to learn from the past trends and apply the patterns to predict future trends or activities. Further, recurrent neural networks (RNN) can be useful in producing relevant results due to their ability to generate an output gained by combining a current input with the outcome from a previous input. The interdependence of the inputs makes it ideal for predicting the results of processes that contain interdependent frames of events. Hence the use of a type of recurrent neural network is recommended for stock market predictions. For the problem discussed above, Long Short Term Memory (LSTM) has been implemented for a higher degree of efficacy. Since LSTM is a modified version of the traditional recurrent neural network, it provides additional advantages over a basic feed-forward approach. It utilizes back-propagation to train the models hence resolving the vanishing gradient problem encountered while using simple RNNs. Also, the changes in stock market prices are heavily time-sensitive and effervescent, making LSTM the best suited for the processing, classification, and prediction of time series that include uncertain time lags. This paper will further delineate some of the existing strategies and related experiments performed to gain ideal results.

III. RELATED WORKS

Since stock market prediction is prominent among researchers and scholars for technical analysis, many techniques have been executed to gather the most robust solutions. Artificial Neural Networks (ANNs) paired with Adaptive Neuro-Fuzzy Inference Systems were applied to stock prices related to various stock exchange cases; various numerical parameters like highest price, lowest price, number of shares traded and the opening price were used to predict the Dhaka Stock Exchange (DSE) data [3]. In [4], Support Vector Regression (SVR) was used, an SVR builds a regression model based on historical time series, and it was concluded that SVR is a reliable tool for predicting stock market prices. Deep learning is also a powerful method for event-driven stock prediction; in [5], novel neural tensor networks and deep convolutional neural networks were implemented to influence the stock market's changing events. Further, fundamental analysis techniques such as textual information and numerical data were used in [6] to explore and classify the textual information derived from news articles using the paragraph vector and utilizing LSTM series-based prediction. Most of the existing experimentation and research work describes the machine learning strategies based on neural networks and regression, but [7] explores the usage of sentiment analysis on the users available on the web and to accomplish this, message boards were primarily targeted to retrieve and classify the messages relevant to the stock exchange. Another paper discussed using Twitter to analyze the sentiments of the users on the social media platform who posted content related to stock market trends [8]. The study presented in [9] has applied a unique approach of working with a combination of wavelet transforms and recurrent neural networks to forecasting stock prices and increasing investments. This paper further optimized the solution by applying the Artificial Bee Colony algorithm on the outcomes of the RNN. The application of LSTM, associated neural networks and deep recurrent neural network model explained to gain the necessary predictions described in [10] provided an inference that suggested that the associated neural networks produce the most precise results. Following the literature reviews on the various models used for prediction, it was decided to understand and practically implement LSTM to predict the opening and the closing price.

IV. METHODS AND ALGORITHMS

A. MODEL USED

Long Short Term Memory Network (LSTM):

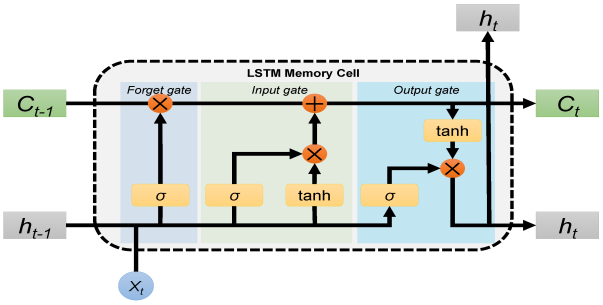

Figure 3: LSTM unit structure

LSTM is a modified version of RNN where every hidden unit is replaced by an LSTM cell, and every cell has a cell state (C_t). As discussed earlier, it has a significant advantage of remembering or forgetting selective patterns for a long time, eliminating the problem of long-term dependency. This is because of its gated structure that is shown in Figure 3. An LSTM unit cell is constructed of three gates: forget gate, input gate, outer gate. The forget gate is responsible for selectively discarding some information. The input gate counts on the amount of information added to the current state, and the outer gate gives the filtered output to the next state. These gates contain a sigmoid function and a dot multiplication operator. The volume of information passed through the gate is decided by a sigmoid layer, which yields an output between 0 and 1. The cell state contains the memory from the previous state to the next state which is also called Long term memory. As seen in Figure 3, C_{t-1} represents the previous state and C_t represents the current state of the cell. Firstly, the forget gate receives the output of the previous state, h_{t-1} and input of current state, x_t, passes it through the sigmoid function to give the Eq. 1,

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right)$$

where W_f is the weight and b_f is the bias. Now, the input gate determines how much new data is to be added to the current state. It is done in two parts. In the first part, information passes through the sigmoid layer and determines which information is to be updated. This gives i_t, (Eq 2):

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right)$$

Then, a new candidate value vector $\hat{C}_t$ is created by tan h activation layer. This gives, (Eq 3)

$$\hat{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right)$$

On combining these two equations, we get (Eq 4),

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

Finally, the cell will give output to the next state. It also passes through two layers. Firstly, the information is passed through the sigmoid layer to get a filtered result and then through the tan h layer which levels the output value in the range -1 to 1. Then at last, we do dot multiplication of both to get the final output (Eq 5).

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = O_t * \tanh\left(C_t\right)$$

B. ERROR ANALYSIS AND PARAMETER SETTING

Mean Square Error:

For the improved accuracy and to determine the error found, Mean Square Error function is applied (MSE). It is the most widely used error computation techniques used in many regression problems. As the name suggests, it is the square of the difference between the actual (computed) value of a parameter to that of the expected value. It predicts the degree to which a deviation occurs between these two values. The mathematical equation used to compute the value of MSE is given below (Eq 6):

$$\mathrm{MSE}\,(y, y') = \frac{\sum_{i=1}^{n} \left(y_i - y_i'\right)^2}{n}$$

In this equation, y_i is the actual or observed value and the y_i' is the predicted value. And the difference is divided by the total number of data points which is represented by 'n'. The aim is to have the least possible value of MSE that would suggest that predicted results and the actual results are similar, hence indicating a higher degree of accuracy.

Step size and Number of Epochs:

In order to perform a graphical representation of the computed results, it is required that certain parameters are set and followed. Moreover, the selected value of the parameters may also determine the performance of the model, eventually influencing the accuracy degree. For the project, batch size and the number of epochs are the two parameters used to derive results. To elaborate, batch size is defined as the number of raining examples that are present at a single time; since the entire data set cannot be used as whole hence it is divided into smaller parts or sets. Number of epochs is the number of times a given dataset passes through a neural network. For the purpose of experimentation, the batch size of 16 has been chosen and 10, 20, and 50 are the epoch values. During experimentation, it was found that the epoch size of 50 gives the least average loss value and results in the highest accuracy of results. In Figure 4, it can be viewed that as the epoch size increases, the accuracy also increases.

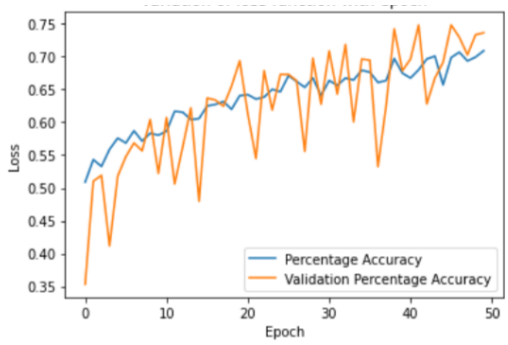

Figure 4 Variation in the accuracy with epoch

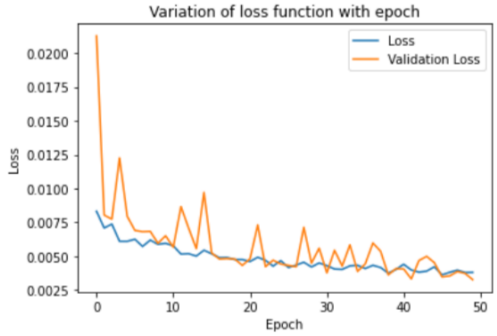

Figure 5. Variation in the loss function with epoch

V. EXPERIMENTAL ANALYSIS AND DISCUSSION

This section of the paper gives a system overview and the following sub-parts offer a detailed view of the experimental setup through the discussion and analysis of the obtained results.

A. DATASET

The dataset used in this project is taken from Yahoo Finance API named yfinance. We fetched the historical price data of smartphone and other gadgets manufacturing company Apple Inc. It has almost 3 thousand records of the last 10 years having the following attributes.

1) Open - Opening price of the stock on a particular day.
2) Close- Closing price of the stock on a particular day.
3) Low- Lowest price of the stock on a particular day.
4) High- Highest price of the stock on a particular day.
5) Volume- It is the total number of shares traded on a particular day.
6) Date- It represents the date.

The date attribute has been considered as the index in our dataset. Further, the dataset is divided into train and test sets. The train set has first 1700 values and the remaining are added to the test set. All the attributes are normalised to a specific range using the MinMaxScaler function.

	Date	Open	High	Low	Close
Date					
2010-01-04	2010-01-04	7.622500	7.660714	7.585000	7.643214
2010-01-05	2010-01-05	7.664286	7.699643	7.616071	7.656428
2010-01-06	2010-01-06	7.656428	7.686786	7.526786	7.534643
2010-01-07	2010-01-07	7.562500	7.571429	7.466072	7.520714
2010-01-08	2010-01-08	7.510714	7.571429	7.466429	7.570714
...	...	...	...	...	...
2020-04-23	2020-04-23	68.967499	70.437500	68.717499	68.757500
2020-04-24	2020-04-24	69.300003	70.752502	69.250000	70.742500
2020-04-27	2020-04-27	70.449997	71.135002	69.987503	70.792503
2020-04-28	2020-04-28	71.269997	71.457497	69.550003	69.644997
2020-04-29	2020-04-29	71.182503	72.417503	70.972504	71.932503

Figure 6: Dataset

B. EXPERIMENTAL FRAMEWORK AND PERFORMANCE MEASURES

The whole project was performed in the following stages:
Stage 1: Collecting Data
In the first stage, we researched various resources to get a reliable dataset for our project. We have used yfinance python API to fetch the data from Apple Inc. [11] This dataset includes the price history of Apple stocks for the last 10 years. The attributes of this dataset are: Open, Close, Adj Close, Low, High, Volume and Date. Date is the index. The stocks of Apple are quite stable for a particular day, so the opening and closing price is mostly the same over last decade. This can be seen in Figure 7, where blue and orange lines almost coincide. We have mentioned this here to give an idea about data distribution.

Figure 7: Plot of opening and closing price

Stage 2: Data Preprocessing:
The dataset that we have used is already very reliable and flawless. There were no missing values in the data. So we just filtered the data by eliminating a few unnecessary attributes like adjacent close price and then scaled it to a specific range using the MinMaxScaler function from sklearn library. After this filtering, the data is split into train and test sets. The train set has 70 percent of data having 1700 entries and is used to train our model. The test set is used in prediction of the stock prices and has 30 percent of data.
Stage 3: Feature Extraction

In this stage, only required features are extracted from the dataset and rest are eliminated. In this dataset, the extracted features are Open, Close, Low, High, Volume and Date.

Stage 4: Training our LSTM model Now, we will train the model using the train set. The training data is fed to the neural network which makes predictions. We have used the keras python library to create our model. Training model is a sequential model composed of 4 LSTM layers each with a dropout and one dense layer at the last [12]. We have used epoch value 50 to fit our model to the training data.

Stage 5: Fitting the model and making predictions In this stage, the test data is used to make predictions of the future stock prices. Along with this the loss and accuracy is also calculated which turns out to be 73 percent approximately.

C. RESULTS OBTAINED

This section explains the experimental result after implementing the project. We have performed various simulations using different numbers of epochs (10, 20, 50) and parameters. Figure 8 shows the result including loss and accuracy after using 50 epochs. It is observed that accuracy increases as epochs increase. The only limit is system configurations. From Figure 4, it is clear that mean square error decreases as we feed more and more data to our model till the verge of overfitting.

```
Epoch 42/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0040 - Percentage_Accuracy: 0.6910 - val_loss: 0.0033 - val_Percentage_Ac
curacy: 0.7483
Epoch 43/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0039 - Percentage_Accuracy: 0.6871 - val_loss: 0.0047 - val_Percentage_Ac
curacy: 0.6280
Epoch 44/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0041 - Percentage_Accuracy: 0.6905 - val_loss: 0.0050 - val_Percentage_Ac
curacy: 0.6675
Epoch 45/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0043 - Percentage_Accuracy: 0.6503 - val_loss: 0.0045 - val_Percentage_Ac
curacy: 0.6913
Epoch 46/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0034 - Percentage_Accuracy: 0.7210 - val_loss: 0.0035 - val_Percentage_Ac
curacy: 0.7483
Epoch 47/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0037 - Percentage_Accuracy: 0.6998 - val_loss: 0.0035 - val_Percentage_Ac
curacy: 0.7296
Epoch 48/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0041 - Percentage_Accuracy: 0.6671 - val_loss: 0.0039 - val_Percentage_Ac
curacy: 0.7028
Epoch 49/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0035 - Percentage_Accuracy: 0.7329 - val_loss: 0.0037 - val_Percentage_Ac
curacy: 0.7334
Epoch 50/50
84/84 [==============================] - 1s 12ms/step - loss: 0.0039 - Percentage_Accuracy: 0.7013 - val_loss: 0.0033 - val_Percentage_Ac
curacy: 0.7364
```

Figure 8: Result of last 10 epochs

We received the most ideal results when we performed the above number of epochs using four extraction features that were discussed earlier, with an accuracy of 73.6 percent. Figure 9 shows the comparison between actual and predicted values of opening and closing stock prices of Apple over a period of 10 years.

Figure 9: Stock Predictions

D. DISCUSSION

The results revealed that the use of LSTM is an ideal method for predicting stock prices as it is capable of computing meaningful and accurate results from historical data. The empirical data also suggests that the extraction features, error analysis and parameters used for prediction successfully contribute to reducing the Mean Squared Error, loss and present accurate results. Further, we attempted to choose the most optimal epoch size to reduce the chances of underfitting and overfitting while ensuring a feasible processing time. Eventually, after performing training analysis and applying varying values of the deciding parameters, we could obtain a high accuracy percentage for the results. Finally, we received a good accuracy of 73.6 percent. On comparing the results with the existing researches stated in the related works, we can conclude that the project was successful in producing a considerably high percentage, and the results can be attributed to the use of a neural network; a fundamental technique like text and sentiment analysis would give unsatisfactory results as it does not take the previous trends and data into consideration. Although we gained impressive results, we could not increase the parameters for the most optimal value due to limitations regarding computer hardware and processing time. Further, we used a large dataset, but a massive dataset could have resulted in an accuracy percentage of more than 90 percent, but we refrained from that due to time constraints. Overall, the theoretical aspects combined with experiential knowledge produced favourable results.

VI. FUTURE SCOPE AND CONCLUSION

The future work related to the research project can proceed in multiple directions. Firstly, with the combination of textual information or data derived from social media and stock market discussion forums, more precise results can be obtained using Natural Language Processing algorithms. The news articles will facilitate predictions based on current trends and global scenarios. Other types of neural networks can also be used for comparison. For instance, deep recurrent neural networks or associated neural networks can be implemented to compare and determine the optimal neural network-based approach. Further, an event-driven in-depth learning approach can also be utilized for a more complex model.

In this paper, we demonstrated the use of a Long Short Term Memory based neural network to predict stock market prices. We included a prominent dataset into our model to train examples and then predicted the opening and closing price after performing normalization, data extraction and loss calculation. The paper described the model, experimentation framework, and methodology to justify the feasibility and accuracy of the applied machine learning technique. Experimental results indicated that the model used proved to successfully predict the required results with a high degree of accuracy.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted

expression “one of us (R. B. G.) thanks . . .”. Instead, try “R. B. G. thanks. . .”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [**?**]. Refer simply to the reference number, as in [**?**]—do not use “Ref. [**?**]” or “reference [**?**]” except at the beginning of a sentence: “Reference [**?**] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [**?**]. Papers that have been accepted for publication should be cited as “in press” [**?**]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [**?**].

REFERENCES

[1] M. Shell. (2007) IEEEtran homepage. [Online]. Available: http://www.michaelshell.org/tex/ieeetran/

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.