



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: <https://doi.org/10.22214/ijraset.2022.45595>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Algorithmic Trading Stock Price Model

Rohan Sharma¹, Venkatsaigautam Bathina²

^{1,2}Department of Computer Science Engineering, Amity University Mumbai

Abstract: Stock market price prediction is a difficult undertaking that generally requires a lot of human-computer interaction. Traditional batch processing methods cannot be used effectively for stock market analysis due to the linked nature of stock prices. This project presents an online learning technique that employs a recurrent neural network of some sort (RNN) called Long Short-Term Memory (LSTM), which uses stochastic gradient descent to update the weights for individual data points. When compared to existing stock price prediction systems, this will yield more accurate results. With varying sizes of data, the network is trained and evaluated for accuracy, and the results are tallied. A comparison with respect to accuracy is then performed against an Artificial Neural Network. Traditional approaches to securities market analysis and stock value prediction embrace basic analysis, that appearance at a stock's past performance and therefore the general believability of the corporate itself, and applied mathematics analysis, that is exclusively involved with computation and distinguishing patterns available value variation.

Keywords: LSTM, Stochastic Gradient, Genetic Algorithm, Streamlit, ANN, YFinance.

I. INTRODUCTION

Traditional approaches to exchange analysis and stock value prediction embrace basic analysis, that appearance at a stock's past performance and therefore the general credibleness of the corporate itself, and applied math analysis, that is only involved with calculation and distinctive patterns available value variation.

The latter is usually achieved with the assistance of Genetic Algorithms (GA) or Artificial Neural Networks (ANN's), however these fail to capture correlation between stock costs within the style of semipermanent temporal dependencies. Another major issue with exploitation easy ANNs for stock predictions the development of exploding / vanishing gradient, wherever the weights of an oversized network either become overlarge or too little (respectively), drastically swiftness their convergence to the best worth. this can be generally caused by 2 factors: weights area unit initialized at random, and therefore the the} weights nearer to the top of the network also tend to alter plenty quite those at the start.

An alternative approach to exchange analysis is to cut back the spatiality of the input file and apply feature choice algorithms to range a core set of options (such as gross domestic product, oil price, rate of inflation, etc.) that have the best impact on stock costs or currency exchange rates across markets. However, this methodology does not think about semipermanent commerce methods because it fails to require the complete history of trends into account; moreover, there's no provision for outlier detection, exchange value prediction may be a troublesome enterprise that typically needs plenty of human- pc interaction, ancient instruction execution ways can not be used effectively for exchange analysis because of the joined nature of stock costs. It has a tendency to gift a web learning technique that employs a continual neural network of some type (RNN) referred to as Long Short Term Memory (LSTM), that uses random gradient descent to update the weights for individual knowledge points. compared to existing stock value prediction systems, this can yield a lot of correct results.

With variable sizes of knowledge, the network is trained and evaluated for accuracy, and therefore the results area unit tallied. A comparison with relevancy accuracy is then performed against a synthetic Neural Network.

II. LITERATURE REVIEW

Kim et al. [1] combined artificial neural networks (ANN) with genetic algorithms (GAs) to predict stock price indexes using discretized features. Data containing 2928 trading days, from January 1989 to December 1998, were used. Factors and formulas were chosen based on their analysis. In addition to dimensionality reduction, they optimized feature discretization. Using GA to optimize ANNs was one of the strengths of their work. As a starting point, the hidden layer has 12 input features and 12 processing elements. Additionally, the authors focused only on two factors of optimization in the learning process of ANN. Nevertheless, GA was still considered to have great potential for optimizing feature discretization.

According to [2], Qiu and Song developed an artificial neural network model to predict Japanese stock market direction. A hybrid GA-ANN model is presented in this study, which combines genetic algorithms and artificial neural networks.

In paper by Lei et al. [3], Wavelet Neural Networks (WNNs) were exploited to predict stock price movements. RS(Rough set) was also used as an optimization technique for attribute reduction by the author. Stock price trend feature dimensions were reduced using Rough Set. Wavelet Neural Networks were also constructed using it. This work relies on five well-known stock market indices as its dataset. Several stock market indices were used in the evaluation of the model, and the results were convincing and generalizable. Rough Set reduces the computational complexity by optimizing the feature dimension before processing. There was no mention of the model's weaknesses in the discussion, but only parameters were adjusted. Moreover, the model may not perform the same way on a specific stock if applied to indices.

Lee in [4] predicted stock trends using SVM and a hybrid feature selection method. In this study, a subset of the NASDAQ Index in Taiwan Economic Journal Database (TEJD) from 2008 was used. Supported sequential forward search (SSFS) played the role of the wrapper in the feature selection process. This work also offers the advantage of a detailed procedure of parameter adjustment that allows the performance to be evaluated at different parameter values. Model structuring at the primary stage is also heuristic with the clear structure of the feature selection model. There were limitations in the study, such as comparing back-propagation neural networks (BPNNs) only and not other machine learning algorithms.

III. PROPOSED SYSTEM

A. Workflow

The Workflow is as follows:

- 1) Visualization of Historical Data
- 2) Pre-Processing of Data
- 3) Divide the data into Training and Testing
- 4) Modelling
- 5) Deployment

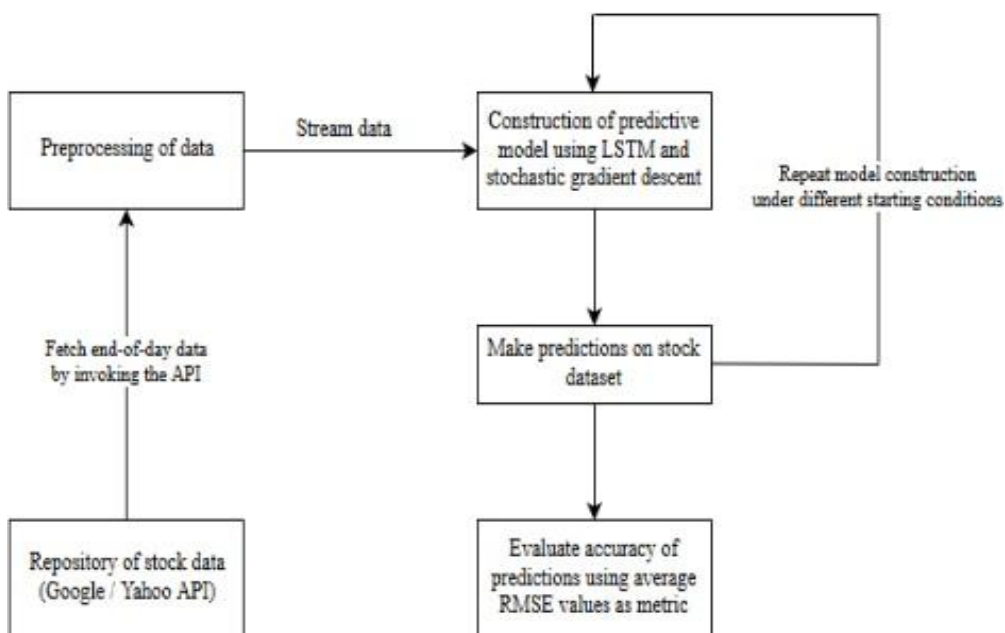


Fig.1 Proposed Model using LSTM and YFinance Library

B. YFinance

Finance came as a support to those who became helpless after the closure of Yahoo Finance historical data API, as many programs that relied on it stopped working. YFinance was created to help the programs and users who were relying on the Yahoo Finance API. It solves the problem by allowing users to download data using python and it has some great features also which makes it favourable to use for stock data analysis. YFinance not only downloads the Stock Price data it also allows one to download all the financial data of a Company since its listing in the stock market. It is easy to use and is blazingly fast. The library is famous for Financial Data Analysis.

C. TensorFlow

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. [5] It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions. TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same. TensorFlow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

D. LSTM

LSTMs are very powerful in sequence prediction problems because they are able to store past information. This is important in this case because the previous price of a stock is crucial in predicting its future price. [6] Using YFinance library to capture the real time or the live data of various companies which will be further used for Modelling.

E. Training and Testing Data

Online and batch learning algorithms differ in the way in which they operate. In an online algorithm, it is possible to stop the optimization process in the middle of a learning run and still train an effective model. This is particularly useful for very large data sets (such as stock price datasets) when the convergence can be measured and learning can be quit early. The stochastic learning paradigm is ideally used for online learning because the model is trained on every data point —each parameter update only uses a single randomly chosen data point, and while oscillations may be observed, the weights eventually converge to the same optimal value. [7] Benchmark stock market data (for end-of-day prices of various ticker symbols i.e., companies) was obtained from one primary source: Yahoo Finance. Yahoo Finance website offer URL- based APIs from which historical stock data for various companies can be obtained for various companies by simply specifying some parameters in the URL.

IV. IMPLEMENTATION DETAILS

After taking the real time or the live data with the help of Y finance library it can start with modelling of the prediction model using LSTM.

```
[ ] import yfinance as yf
import pandas as pd

[ ] df = yf.download('RELIANCE.NS')

[*****100%*****] 1 of 1 completed

[ ] df.head()
```

	Open	High	Low	Close	Adj Close	Volume
Date						
1996-01-01	15.859429	15.944529	15.754989	15.917452	10.509538	48051995
1996-01-02	15.878770	15.956133	15.677626	15.793671	10.427812	77875009
1996-01-03	16.052837	16.783918	15.878770	15.913584	10.506985	96602936
1996-01-04	15.762726	15.813012	15.553845	15.766594	10.409933	100099436
1996-01-05	15.704703	15.704703	15.522900	15.658285	10.338421	76935930

Fig.2 Data Acquisition using YFinance Library

Visualizing the stock price right from the date the company was listed in Stock Exchange either in (NSE / BSE). It helps us in understanding the volatility of company’s stock price and the approximate rate of growth of share price annually which can be further used for various studies in analysing the company’s future in terms of stability.

Once the LSTM model is fit to the training data, it can be used to predict the end-of-day stock price of an stock. This prediction can be performed in two ways:

- 1) *Static*: a simple, less accurate method where the model is fit on all the training data. Each new time step is then predicted one at a time from test data. [8]
- 2) *Dynamic*: a complex, more accurate approach where the model is refit for each time step of the test data as new observations are made available. The accuracy of the prediction model can then be estimated robustly using the RMSE (Root Mean Squared Error) metric. [9] This is due to the fact that neural networks in general (including LSTM) tend to give different results with different starting conditions on the same data. It then repeats the model construction and prediction several times (with different starting conditions) and then take the average RMSE as an indication of how well the configuration would be expected to perform on unseen real-world stock data. That is, it will compare the predictions with actual trends in stock price movement that can be inferred from historical data. [10]

Given below is a brief summary of the various terminologies relating to the proposed stock prediction system: Training set: subsection of the original data that is used to train the neural network model for predicting the output values [11] Test set: part of the original data that is used to make predictions of the output value, which are then compared with the actual values to evaluate the performance of the model Validation set: portion of the original data that is used to tune the parameters of the neural network model. [12] Batch size: number of samples that must be processed by the model before updating the weights of the parameters [13] Epoch: a complete pass through the given dataset by the training algorithm Loss function: a function, defined on a data point, prediction and label, that measures a penalty such as square loss. [14] Root Mean Square Error (RMSE): measure of the difference between values predicted by a model and the values actually observed. It is calculated by taking the summation of the squares of the differences between the predicted value and actual value, and dividing it by the number of samples.

```
[16] model = Sequential()
model.add(LSTM(50,return_sequences = True, input_shape = (x_train.shape[1], 1)))
model.add(LSTM(50,return_sequences = False))
model.add(Dense(25))
model.add(Dense(1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.fit(x_train, y_train, batch_size=40, epochs=20)

Epoch 1/20
127/127 [=====] - 11s 61ms/step - loss: 6.5594e-04
Epoch 2/20
127/127 [=====] - 7s 58ms/step - loss: 5.0681e-05
Epoch 3/20
127/127 [=====] - 7s 59ms/step - loss: 4.8207e-05
Epoch 4/20
127/127 [=====] - 8s 60ms/step - loss: 4.4784e-05
Epoch 5/20
127/127 [=====] - 7s 55ms/step - loss: 4.0820e-05
Epoch 6/20
127/127 [=====] - 7s 55ms/step - loss: 3.7719e-05
Epoch 7/20
127/127 [=====] - 7s 56ms/step - loss: 3.8774e-05
Epoch 8/20
127/127 [=====] - 7s 58ms/step - loss: 3.6472e-05
Epoch 9/20
127/127 [=====] - 7s 59ms/step - loss: 3.6442e-05
Epoch 10/20
127/127 [=====] - 7s 57ms/step - loss: 3.2128e-05
Epoch 11/20
127/127 [=====] - 7s 58ms/step - loss: 3.0201e-05
Epoch 12/20
127/127 [=====] - 7s 58ms/step - loss: 3.0460e-05
Epoch 13/20
127/127 [=====] - 8s 59ms/step - loss: 2.8723e-05
Epoch 14/20
127/127 [=====] - 8s 63ms/step - loss: 2.7039e-05
Epoch 15/20
127/127 [=====] - 8s 61ms/step - loss: 2.5161e-05
Epoch 16/20
127/127 [=====] - 8s 61ms/step - loss: 2.5554e-05
Epoch 17/20
127/127 [=====] - 7s 57ms/step - loss: 2.5606e-05
Epoch 18/20
127/127 [=====] - 7s 57ms/step - loss: 2.3844e-05
Epoch 19/20
127/127 [=====] - 7s 56ms/step - loss: 2.2577e-05
Epoch 20/20
127/127 [=====] - 7s 57ms/step - loss: 2.3445e-05
```

Fig.3 Modelling using LSTM

V. RESULTS AND ANALYSIS

The accuracy of the prediction model can then be estimated robustly using the RMSE (Root Mean Squared Error) metric. This is due to the fact that neural networks in general (including LSTM) tend to give different results with different starting conditions on the same data. Then repeat the model construction and prediction several times (with different starting conditions) and then take the average RMSE as an indication of how well the configuration would be expected to perform on unseen real-world stock data. That is, compare the predictions with actual trends in stock price movement that can be inferred from historical data Fig.4 and Fig.5 shows the actual and the predicted closing

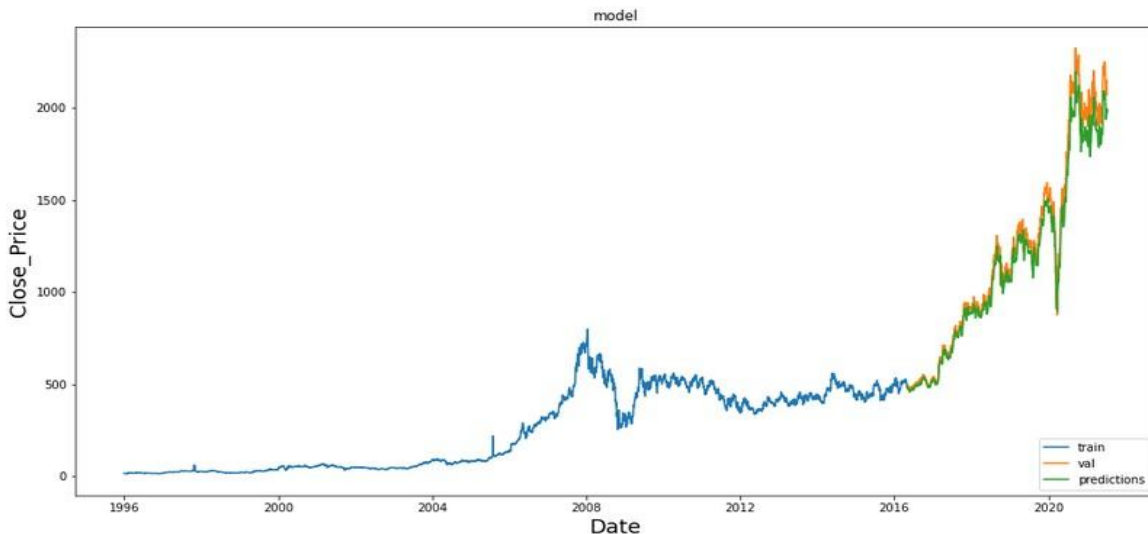


Fig.4 Closing Price with respect to date

price of the company Alcoa Corp, a large-sized stock. The model was trained with a batch size of 40 and 20 epochs, and the predictions made closely matched the actual stock prices, as observed in the graph.

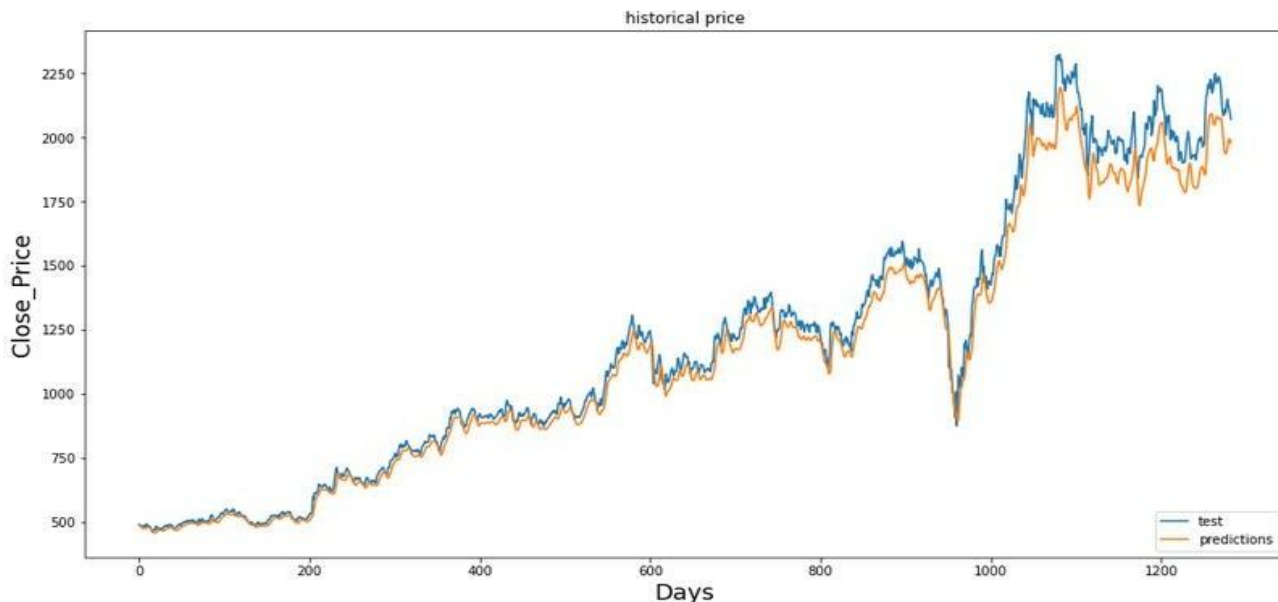


Fig.5 Closing Price with respect to days

Now using streamlit it can be deployed using any machine learning model and any python project with ease and without concerns about the frontend. Streamlit is very user friendly. Streamlit is an open- source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

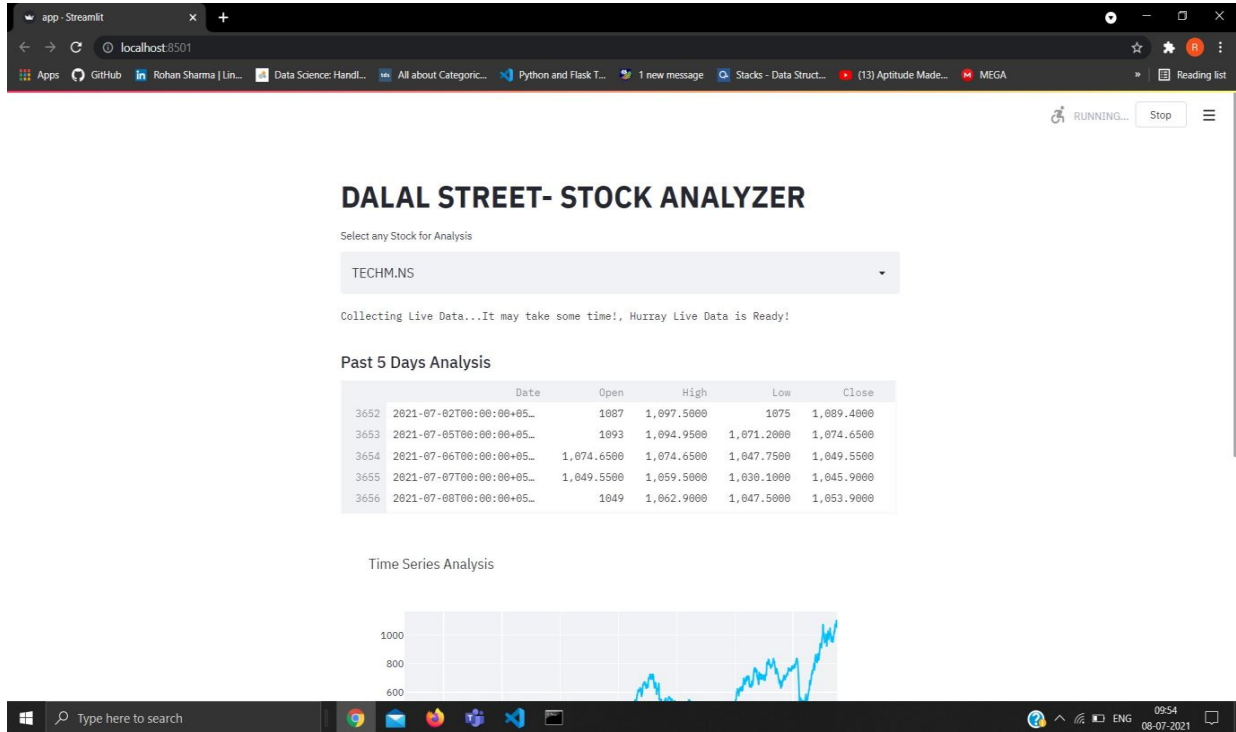


Fig.6 User Interface using Streamlit

A comparison with Plotly Dash probably deserves its own blog post but the Dash approach is very much more in the camp of making React easier to do in Python, for those with experience in traditional GUI application frameworks, Streamlit works more like an immediate mode user interface. That is, it reruns the script from top to bottom each time a UI action (e.g., button is clicked) is performed. Below Fig.7 is the interface of the website deployed on Streamlit platform. User can select any stock of choice from the list of options. After selection of the stock it takes a couple of seconds to capture the latest price with the help of yfinance library.

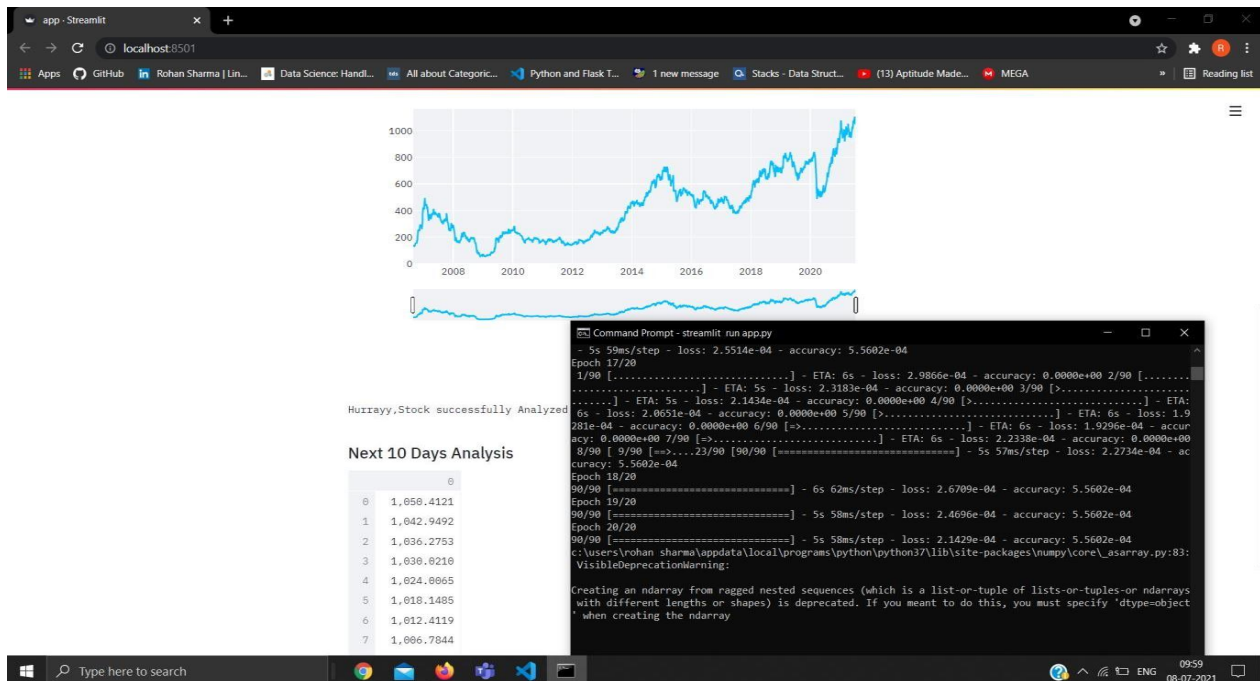


Fig.7 Price Forecasting using LSTM

After capturing the price of that stock it runs the LSTM Model in the backend and make prediction based on batch size and epochs that is considered in the code. Prediction usually takes upto 2-3 mins. Prediction are made for the next 10 days only (considering the high volatility of Stock Market).

VI. CONCLUSION

The Stock markets are hard to monitor and require plenty of context when trying to interpret the movement and predict prices. [15] In ANN, each hidden node is simply a node with a single activation function, while in LSTM, each node is a memory cell that can store contextual information. As such, LSTMs perform better as they are able to keep track of the context-specific temporal dependencies between stock prices for a longer period of time while performing predictions.

An analysis of the results also indicates that model gives better accuracy when the size of the dataset increases. With more data, more patterns can be fleshed out by the model, and the weights of the layers can be better adjusted. At its core, the stock market is a reflection of human emotions. Pure number crunching and analysis have their limitations; a possible extension of this stock prediction system would be to augment it with a news feed analysis from social media platforms such as Twitter, where emotions are gauged from the articles. [16]. This sentiment analysis can be linked with the LSTM to better train weights and further improve accuracy.

REFERENCES

- [1] Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl.* 2000;19:125–32. [https://doi.org/10.1016/S0957-4174\(00\)00027-0](https://doi.org/10.1016/S0957-4174(00)00027-0).
- [2] Qiu M, Song Y. Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE.* 2016;11(5):e0155133.
- [3] Lei L. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Appl Soft Comput J.* 2018;62:923–32. <https://doi.org/10.1016/j.asoc.2017.09.029>.
- [4] Lee MC. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Syst Appl.* 2009;36(8):10896–904. <https://doi.org/10.1016/j.eswa.2009.02.038>.
- [5] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [6] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. *Lecture Notes in Statistics*. Berlin, Germany: Springer, 1989, vol. 61.
- [7] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [8] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, “High resolution fiber distributed measurements with coherent OFDR,” in *Proc. ECOC’00*, 2000, paper 11.3.4, p. 109.
- [9] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.
- [10] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [11] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>
- [12] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.
- [13] “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.
- [14] A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [15] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” *Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep.* 99-02, 1999.
- [16] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)