© INNOVATIVE RESEARCH THOUGHTS
Refereed | Peer Reviewed | Indexed
ISSN : 2454 – 308X | Volume : 09 , Issue : 01 | January - March 2023

STOCKIFY

SANJANA BHOLA
Deprt. of Business Administration
GGDSD College
Chandighar
sanjanabhola3112@gmail.com

SHREYANSH KUMAR
Deprt. of Computer Science
Bennett University
Greater Noida
shreyanshkumar2304@gmail.com

RONAK SINGHATWADIA
Deprt. of Computer Engineering
MPSTME
Mumbai, India
ronak.singhatwadia04@nmims.edu.in

DOI : 10.5281/zenodo.7638719

***Abstract*—**
Predicting stock value is a difficult task that ne-cessitates a solid computational foundation in order to compute longer-term share values. Because stock prices are correlated by nature of the market, it will be impossible to estimate costs. The proposed algorithm uses market data to predict share price usingmachine learning techniques such as recurrent neural networks with Long Short Term Memory, with weights modified for each data point using stochastic gradient descent. In compared to currently available stock price predictor algorithms, this method will deliver reliable results. To encourage the graphical outcomes,the network is trained and assessed with varied sizes of input data.

***Index Terms*—Recurrent Neural Network; Long Short-Term Memory; Stock Market; forecasting; prediction**

I. INTRODUCTION

The stock market is made up of a large number of investors and traders who purchase and sell stocks, causing prices to rise or fall. The principles of demand and supply determine stock prices, and the ultimate purpose of stock purchases is to profit by purchasing stocks in firms whose perceived worth (i.e., share price) is projected to rise. The rise and fall of stock prices can be traced back to various Key Performance Indicators (KPIs), which are directly tied to the realm of economics. The initial stock price ('Open'), end-of-day price ('Close'), intraday low price ('Low'), intraday peak price ('High'), and total volume of stocks traded during the day ('Volume') are the five most regularly utilised KPIs. Subjective impressions of the stock market are largely responsible for economics and stock prices. Due to the unpredictability of elements that influence price movement, it is nearly difficult to anticipate stock values to the penny. It is, nonetheless, possible to make an educated pricing estimate. Stock prices never fluctuate in isolation: one stock's change tends to have an avalanche impact on several others. This element of stock price movement can be used to predict the prices of multiple equities at the same time.

Because of the large amount of money involved and the number of transactions that occur every minute, there is a trade-off between prediction accuracy and volume. As a result,the majority of stock prediction systems are distributed and parallelized. These are some of the factors to consider and difficulties to overcome in stock market analysis.

II. LITERATURE SURVEY

Our literature review's initial goal was to look into generic online learning algorithms and see whether they could be applied to our use case, which was working with real-time stock price data. We decided to look at the existing systems, examine the primary shortcomings, and see if we could improve upon them because we couldn't identify any feasible adaptations of these for stock price prediction. The major issuethat we wanted to solve was the correlation between stock data(in the form of dynamic, long-term temporal interdependence between stock prices). RNNs and LSTM were discovered aftera quick search for generic answers to the aforesaid problem.

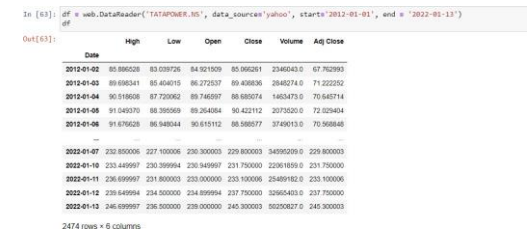
III. DRAWBACKS OF EXISTING SYSTEMS

Fundamental analysis, which looks at a stock's past performance and the company's overall credibility, and statistical analysis, which is solely concerned with number crunching and identifying patterns in stock price variation, are two traditional approaches to stock market analysis and stock price prediction. The latter is often accomplished with the help of Genetic Algorithms (GA) or Artificial Neural Networks (ANN), although these fail to capture long-term temporal dependencies in stock values. Another big concern with utilising simple ANNs for stock prediction is the exploding / disappearing gradient phenomena, which occurs when the weights of a large network become either too large or too tiny (respectively), significantly reducing their convergence to the ideal value. This is due to two factors: weights are started at random, and weights at the end of the network tend to change much more than those near the beginning. Shortlisting a core collectionof features (such as GDP, oil price, inflation rate, etc.) that have the greatest impact on stock prices or currency exchange rates across markets is an alternate way to stock market analysis. However, because it does not take into account the whole history of trends, this method does not address long- term trading strategies; also, there is no provision for outlier detection.

IV. LSTM OVERVIEW

LSTMs are a type of RNN that may capture context-specific temporal dependencies over lengthy time periods. Each LSTM neuron is a memory cell that may store other data and keeps track of its own cell state. An LSTM neuron additionally takes in its old cell state and outputs its new cell state, whereas neurons in standard RNNs only take in their prior hidden state and the current input to output a new hidden state.

© INNOVATIVE RESEARCH THOUGHTS
Refereed | Peer Reviewed | Indexed
ISSN : 2454 – 308X | Volume : 09 , Issue : 01 | January - March 2023



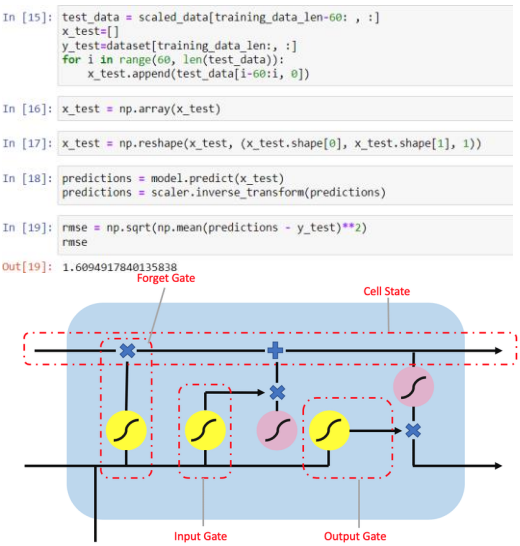


Fig. 1. LSTM Structure

An LSTM memory cell, as depicted in Figure, has the following three components, or gates:

1. Forget gate: The forget gate determines when particular elements of the cell state should be replaced with something else. Information that is current It produces values that are close to one. if there are aspects of the cell state that should be

2. kept, and for values that should be ignored, use zero.

3. Input gate: This portion of the network learns the conditions under which any information should be stored (or updated) in the cell state based on the input (i.e., previous output o(t-1), input x(t), and prior cell state c(t-1).

4. Output gate: This section determines what information is transmitted forward (i.e., output o(t) and cell status c(t)) to the next node in the network based on the input and cell state.

As a result, LSTM networks are suitable for examininghow price changes in one stock affect the prices of numerous other stocks over time. They can also select (in real time) howlong information about certain previous trends in stock price movement should be preserved in order to better anticipate future trends in stock price variance.

V. METHODOLOGY

This section we will discuss the methodology of our system. Our system consists of several stages which are as follows:-

A. Raw Data

In this stage, the historical stock data is collected and this historical data is used for the prediction of future stock prices. Data Collection is a crucial step as it fetches our Dataset for the Stock prices of a company. We use Yahoo Finance as our Data Source. It has proved reliable and useful when deriving data from various national as well as International Companies. Basically, it provides us with the stock quote.

Fig. 2. Reading Stock Data from Data Source

Data Preprocessing Feature Extraction

The pre-processing stage includes the following steps: a) Data discretization: This is a type of data reduction that is especially important for numerical data. Normalization is a type of data transformation. b) Clean up the data: Replaceany missing values. d) Data integration: This is the processof combining data files. To evaluate, the dataset is dividedinto training and testing sets once it has been turned into a clean dataset. The training values are used as the most recent values in this case. Testing data accounts for 5-10 percent of the whole dataset.

```
In [65]: data = df.filter(['Close'])
         dataset = data.values
         training_data_len = math.ceil(len(dataset)*.8)
         training_data_len

Out[65]: 1980

In [66]: scaler = MinMaxScaler(feature_range=(0,1))
         scaled_data = scaler.fit_transform(dataset)
         scaled_data

Out[66]: array([[0.25115767],
                [0.27003844],
                [0.26689164],
                ...,
                [0.89478268],
                [0.91500005],
                [0.94782615]])
```

Fig. 3. Filtering, Scaling, and Transformation of features

B. Constructing the model

The first LSTM layer uses 50 neurons and needs a defined input shape as its the first layer and a TRUE return Sequence as we have another LSTM Layer. The second LSTM layer alsohas 50 neurons assigned. We add two Dense layers totaling 26 neurons assigned. The dense layer is the most commonly used layer for ML models, this is because all of the previous layers provide the Dense layer with an input. It is a deeply connected neural network layer. LSTMs are capable of learning long-term dependencies in data.

© INNOVATIVE RESEARCH THOUGHTS
Refereed | Peer Reviewed | Indexed
ISSN : 2454 – 308X | Volume : 09 , Issue : 01 | January - March 2023

Fig. 4. Creating the Model using different layers

C. Training the model

In this stage, the data is fed to the neural network and trained for prediction assigning random biases and weights.

D. Evaluation

We scale the predicted values up, then we compute the root mean squared error. The RMSE is an indicator that shows us how far apart our predicted values are from the expected value. This is used at the end of a prediction, where the model will spit out a normalized number between 0 and 1, we want to apply the reverse of the dataset normalization to scale it back up to real-world values. Thereby finding our predicted value.

Fig. 5. Reverse Transformation and RMSE Calculation

VI. RESULTS

To put the model to the test, we predict the closing price of tata power. We apply the model to the last 60 days data of the stock and then predict its closing price.

```
In [22]: ttp_quote = web.DataReader('TATAPOWER.NS', data_source='yahoo',start='2012-01-01', end='2022-02-16')
         new_df = ttp_quote.filter(['Close'])
         last_60_days = new_df[-60:].values
         last_60_days_scaled = scaler.transform(last_60_days)
         X_test=[]
         X_test.append(last_60_days_scaled)
         X_test = np.array(X_test)
         X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
         pred_price = model.predict(X_test)
         pred_price = scaler.inverse_transform(pred_price)
         print(pred_price)

         [[231.13066]]

In [23]: ttp_quote2 = web.DataReader('TATAPOWER.NS', data_source='yahoo',start='2022-02-10', end='2022-02-18')
         print(ttp_quote2['Close'])

         Date
         2022-02-10    241.300003
         2022-02-11    232.300003
         2022-02-14    220.500000
         2022-02-15    230.399994
         2022-02-16    231.350006
         2022-02-17    227.750000
         2022-02-18    228.800003
         Name: Close, dtype: float64
```

Fig. 6. Predicting the close price

We then visualize the train and predicted values by plotting it on a graph. This will provide us better insight on how well the model has performed.

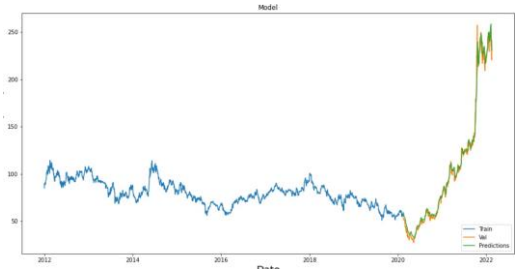


Fig. 7. Visualizing the train and predicted values

```
In [12]: 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))

In [13]: model.compile(optimizer='adam',loss='mean_squared_error')

In [14]: model.fit(x_train, y_train, batch_size=1, epochs=1)
         1938/1938 [==============================] - 35s 17ms/step - loss: 3.5668e-04
```

For a better comparison, we list the closing price and the predicted price. This shows that our model has performedvery well making use of LSTM.

```
In [21]: valid
```

Out[21]:

Date	Close	Predictions
2020-02-14	53.799999	59.220573
2020-02-17	51.750000	58.427197
2020-02-18	51.549999	57.202610
2020-02-19	51.349998	56.007767
2020-02-20	51.599998	55.049221
...	...	...
2022-02-10	241.300003	246.704926
2022-02-11	232.300003	243.607697
2022-02-14	220.500000	239.664001
2022-02-15	230.399994	233.113983
2022-02-16	231.350006	230.757828

Fig. 8. List of close price and predicted price

VII. FRONTEND DEPLOYMENT

It is really neccessary to deploy the model in a way that makes it easy for the user to interact with it without worrying about the backend side of the work. This also aids in testing the model and allows the developers to efficiently deploy the model. So, we have deployed our model to the frontend using technologies like ionic, angular, nodejs and flask.

Fig. 9. Home page

Fig. 10. News page

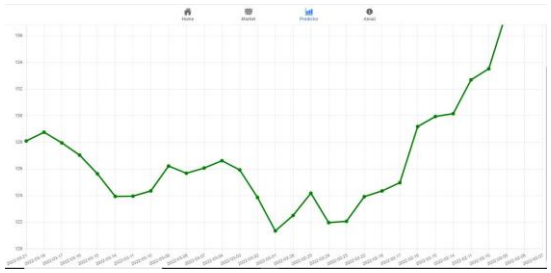


Fig. 11. Prediction page

VIII. CONCLUSION

The popularity of stock market trading is continuously increasing, prompting experts to develop new methods for predicting the future utilising new techniques. The forecasting technique benefits not only scholars, but also investors and anybody who works in the stock market. A forecasting model with high accuracy is required to assist in the prediction of stock indexes. We used one of the most exact forecasting technologies in this study, the Long Short-Term Memory unit, which aids investors, analysts, and anybody interested in investing in the stock market by providing them with a good understanding of the stock market's future position.

IX. FUTURE WORK

The stock market is, at its foundation, a reflection of human emotions. Pure numerical crunching and analysis has its limitations; a potential development of this stock prediction system would be to combine it with a news feed analysis from social media platforms like Twitter, where emotions are gauged from the articles. This sentiment analysis can be combined with the LSTM to increase weight training and accuracy. In the future, we plan to add sentiment analysis from social media to our programme to better understand what the market thinks about price variations for specific stocks. We can do this by adding Twitter and Facebook APIs to our programme, as Facebook is a popular social media platform with a lot of market trend information posted by users.

REFERENCES

[1] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 544-547, doi: 10.1109/AIAM48774.2019.00113.

[2] K. J, H. E, M. S. Jacob and D. R, "Stock Price Prediction Based on LSTM Deep Learning Model," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021, pp. 1-4, doi: 10.1109/ICSCAN53069.2021.9526491.

[3] S. B. Islam, M. M. Hasan and M. M. Khan, "Prediction of Stock Market Using Recurrent Neural Network," 2021 IEEE 12th An- nual Information Technology, Electronics and Mobile Communica- tion Conference (IEMCON), 2021, pp. 0479-0483, doi: 10.1109/IEM-CON53756.2021.9623206.

[4] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 544-547, doi: 10.1109/AIAM48774.2019.00113.

[5] W. Yang, R. Wang and B. Wang, "Detection of Anomaly Stock Price Based on Time Series Deep Learning Models," 2020 Management Sci- ence Informatization and Economic Innovation Development Conference (MSIEID), 2020, pp. 110-114, doi: 10.1109/MSIEID52046.2020.00029.

[6] Y. ZHANG and S. YANG, "Prediction on the Highest Price of the Stock Based on PSO-LSTM Neural Network," 20193rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), 2019, pp. 1565-1569, doi: 10.1109/EITCE47263.2019.9094982.

[7] M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi- Directional LSTM Model," 2020 2nd Novel Intelligent and Lead- ing Emerging Sciences Conference (NILES), 2020, pp. 87-92, doi: 10.1109/NILES50944.2020.9257950.

[8] J. Du, Q. Liu, K. Chen and J. Wang, "Forecasting stock pricesin two ways based on LSTM neural network," 2019 IEEE 3rd In-formation Technology, Networking, Electronic and Automation Con- trol Conference (ITNEC), 2019, pp. 1083-1086, doi: 10.1109/IT-NEC.2019.8729026.

[9] V. Karlis, K. Lepenioti, A. Bousdekis and G. Mentzas, "Stock Trend Prediction by Fusing Prices and Indices with LSTM Neu-ral Networks," 2021 12th International Conference on Information, Intelligence, Systems Applications (IISA), 2021, pp. 1-7, doi: 10.1109/IISA52424.2021.9555506.

[10] S. Mootha, S. Sridhar, R. Seetharaman and S. Chitrakala, "Stock Price Prediction using Bi-Directional LSTM based Sequence to Se-quence Modeling and Multitask Learning," 2020 11th IEEE An- nual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 2020, pp. 0078-0086, doi: 10.1109/UEM-CON51285.2020.9298066.